2023 IEEE International Conference on Multimedia and Expo (ICME) | 978-1-6654-6891-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICME55011.2023.00252

POST-TRAINING QUANTIZATION FOR VISION TRANSFORMER IN TRANSFORMED DOMAIN

Kai Feng[1,2], *Zhuo Chen*[1*], *Fei Gao*[3,4], *Zhe Wang*[5], *Long Xu*[3], *Weisi Lin*[2,5]

1. Peng Cheng Laboratory, Shenzhen, China
2. China-Singapore International Joint Research Institute (CSIJRI), Guangzhou, China
3. National Space Science Center, Chinese Academy of Sciences, Beijing, China
4. University of Chinese Academy of Sciences, Beijing, China
5. Nanyang Technological University, Singapore

ABSTRACT

As a successor to convolutional neural networks (CNNs), transformer-based models have achieved great performance in computer vision tasks. Compressing vision transformers to low-bit brings a number of practical benefits, including higher inference speed, improved memory footprint, and reduced energy consumption. Existing model compression methods, especially quantization techniques, ignore the joint statistics of weights, resulting in sub-optimal task performance at a given quantization bit rate. In this paper, we propose to apply a transform before quantization to decorrelate vision transformer's weights. And the entire compression flow is optimized in a rate-distortion framework to minimize the network output errors instead of simply optimizing for quantization errors or layer-wise output errors. Extensive experimental results on a variety of vision transformers (e.g. Swin, ViT and DeiT) demonstrate that our proposed method outperforms the state-of-the-art. It can quantize vision transformers (e.g. Swin, ViT and DeiT) on both weights and activations to 6-bit without a significant accuracy drop.

Index Terms— Vision transformer, post-training quantization, transform, model compression

1. INTRODUCTION

Following the success in natural language processing(NLP) tasks, transformer-based deep learning models have demonstrated great advantages in computer vision tasks, such as image classification [1], object detection [2], and image super resolution [3]. For instance, in image classification, conventional CNN models, such as VGGNet and ResNet, can achieve 71.62% and 78.63% top-1 accuracy respectively on ImageNet, while the transformer-based model, like ViT-L, can reach 85.63%. However, the vision transformer's great task performance also comes with hundreds of millions of parameters and high computational complexity. The ViT-L model has 307M parameters and takes over 116G FLOPs to classify a 224×224 image, while ResNet only has 60M parameters and takes 11.58G FLOPs. On edge devices with severe resource constraints, such a huge amount of parameters and computational complexity will seriously hinder the deployment of vision transformers. In view of this, it is meaningful to develop model compression techniques for vision transformers.

Recent deep learning model compression techniques can be generally categorized into pruning [4], knowledge distillation [5], low rank approximation [6] and quantization [7]. Among them, quantization approaches have shown great effectiveness and versatility, which can achieve very low bitrate at various rate-accuracy trade-offs for either post-training or re-training. In addition, quantization do not require to change neural network structures, which great fits for well-designed or well-trained models. However, most quantization methods in literature are designed for CNNs, but only a few for transformers. Prior to this work, PTQ for ViT [8], PSAQ [9], FQ-ViT [10], PTQ4ViT [11] proposed post-training quantization methods for vision transformers. However, in these works, quantizers are only optimized for quantization errors or layer-wise output errors. In addition, joint statistics of weights are not explored, which may result in sub-optimal results.

In this paper, we propose to quantize vision transformer in the transformed domain. The proposed method derives proper transforms before quantization to decorrelate the transformer's weights, so that simple scalar quantizer can efficiently represent each weight independently. Furthermore, we formulate the compression process as a rate-distortion optimization (RDO) problem, which is directly optimized for neural network output error maintaining good task performance after quantization. Main contributions of this paper are summarized as follows.

- We propose a transform quantization method for the vision transformer compression on both weights and ac-

*Corresponding author: Zhuo Chen (chenzhuo.zoom@gmail.com)

This work was supported in part by the Basic and Frontier Research Project of PCL, in part by Major Key Project of PCL, in part by the National Key R&D Program of China (No. 2021YFA1600504), and in part by the National Natural Science Foundation of China(NSFC) (No. 11790305).

1945-788X/23/$31.00 ©2023 IEEE
DOI 10.1109/ICME55011.2023.00252

tivations. In transformed domain, scalar quantizer can better represent each weight/activation element independently.

- We model the quantization process as a rate-distortion optimization problem, minimizing the network output error instead of quantization error or layer-wise output error, which largely maintains the task performance after quantization.
- The proposed method outperforms state-of-the-art methods on a broad range of vision transformers(such as Swin, ViT, and DeiT). It can compress the vision transformer down to 6-bit without a noteworthy accuracy drop ($< 0.5\%$).

2. RELATED WORK

2.1. Vision transformer

In recent years, vision transformers obtained unparalleled performance in a broad range of computer vision tasks. In image classification [1], transformer took 1-D sequences instead of 2-D images as the input, which outperformed CNNs. In image segmentation [12], a transformer-based backbone was employed instead of traditional U-Net to segment medical images. YOLOS [13] made the attempt to perform object detection in a fully sequence-to-sequence fashion using only pre-trained ViT encoders. [14] proposed a spatial-temporal ViT to apply transformer in image as well as video super-resolution. [15] employed a new ViT model for image anomaly detection and location (VT-ADL). In low level vision tasks, transformer has also been approved to be effective for image denoising [16].

Transformers outperform CNNs in computer vision tasks with disparate network structures. The key component of CNNs is the convolution layer, whose weights are usually in form of 3-dimensional convolution kernel. On the contrary, transformers largely consist of MSA and MLP modules, whose weights are basically matrices. In view of the different structural properties, compression methods of CNN cannot naively be applied for transformer. Compression for vision transformer has become an emerging topic lately.

2.2. Post-training quantization for vision transformer

Deep learning quantization methods can generally be categorized into two types: Post-Training Quantization (PTQ) [17] and Quantization-Aware Training (QAT) [18]. QAT methods help to compensate the quantization error by re-training the deep model after quantization, so that to obtain quantized network with low precision drop. But the re-training process requires training set, long optimization time, and hyper-parameters tuning, which makes QAT not the best choice when labelled data set is not available or rapid deployment is required. On the contrary, PTQ methods only use unlabeled calibration images to quantify the network, where the compressed model can rapidly be obtained on top of the well-trained full-precision one.

In the literature, there are a few attempts to perform post-training quantization on vision transformers. PTQ for ViT [8] quantized transformer's weights and activations with the optimization goal of minimizing the activations quantization error measured by Pearson correlation coefficient and maintaining the relative order of self-attention. On top of this work, FQ-ViT [10] proposed Log-Int-Softmax(LIS) and Power-of-Two Factor(PTF) to further quantize the LayerNorm and Softmax. The proposed methods determined the optimal quantization intervals by considering the MSE between the pristine and reconstructed data. PTQ4ViT [11] employed the Hessian guided metric to represent the distance between pristine and reconstructed intermediate layer outputs. Scaling factor for the quantizer is optimized to minimize such distance. PSAQ [9] proposed a data-free quantization framework which generates input samples from Gaussian noise, and then solves the quantization parameter with the synthetic samples. All the aforementioned post-training quantization methods are optimized for the quantization errors or layer-wise output distance, and none of them perform quantization from the perspective of the transform domain. They can generally quantize the vision transformers to 8-bit without significant accuracy drop ($< 0.5\%$), which can be seen in Table 1.

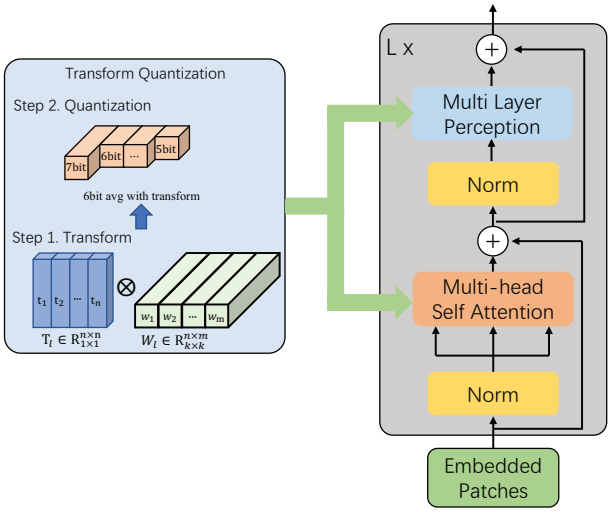

Fig. 1: Quantization for ViT in transformed domain. A transform is applied to the weights in MSA and MLP, following with a naive clipped scalar quantizer with optimized bit allocation.

3. METHODOLOGY

In this section, the proposed transform quantization method for the vision transformer is introduced in detail. The weights and activations in the Multi-Head Self Attention (MSA) and Multi-Layer Perceptron (MLP) are quantized. Fig. 1 shows the compression flow of the proposed method. Before quantizing the weights in the transformed domain, a proper trans-

form is first applied to the weights. Then a naive clipped scalar quantizer can be performed on the transformed weights with mixed precision which is controlled by a bit allocation algorithm. In particular, subsections 3.3 and 3.4 will elaborate how the transform and the bit allocation are determined.

3.1. Preliminaries

Input of a transformer is usually the 1-D sequence of token embeddings. In order to process 2-D images $x \in \mathbb{R}^{H\times W\times C}$, the vision transformer reshapes the input images into a sequence of flattened 2D patches $x_p \in \mathbb{R}^{N\times(P^2C)}$, where H and W are the resolutions of the original image, C is the number of channels and (P, P) is the resolution of each image patch, and $N = \frac{HW}{P^2}$ is the number of patches, which determines input sequence length for the vision transformer. Same as transformer in NLP, the vision transformer maps each patch to a vector by a trainable linear projection. Due to the constant widths over all the layers, the vision transformer takes the input of its first layer as:

$$x_0 = [x_{class}; x_p^1\mathbf{W}; x_p^2\mathbf{W}; ...; x_p^N\mathbf{W};] + \mathbf{E}_{pos} \quad (1)$$

where $\mathbf{W} \in \mathbb{R}^{(P^2\cdot C)\times D}$, $\mathbf{E}_{pos} \in \mathbb{R}^{(N+1)\times D}$.

For the Multi-Head Self Attention (MSA) module, let Q, K and V denote query, key, and value respectively. The input of the softmax function is the similarity between Q and K by calculating their dot product. Output of MSA is the attention map, which is the product of V and the output of softmax layer. This process can be formulated as:

$$\mathrm{MSA}\,(x_l) = \mathrm{Softmax}\left(\frac{Q_l K_l^T}{\sqrt{D}}\right) V\mathbf{W}_l^{msa} \quad (2)$$

$$\text{where } Q_l = x_l\mathbf{W}_l^q, K_l = x_l\mathbf{W}_l^k, V_l = x_l\mathbf{W}_l^v \quad (3)$$

For the Multi-Layer Perceptron (MLP) module which consists of two linear layers and a GELU layer contains parameters $\mathbf{W}^1 \in \mathbb{R}^{D\times D_h}$, $\mathbf{W}^2 \in \mathbb{R}^{D_h\times D}$ and $b \in \mathbb{R}^D$, the output can be expressed as:

$$\mathrm{MLA}(x_l) = \mathrm{GELU}(x_l\mathbf{W^1} + \mathbf{b}^1)\mathbf{W^2} + \mathbf{b}^2 \quad (4)$$

A typical l-th vision transformer encoder consisting of MSA and MLP can be expressed by combining Equation 2 and Equation 4 as:

$$x_l' = \mathrm{MSA}(\mathrm{LN}(x_l)) + x_{l-1} \quad (5)$$

$$x_l = \mathrm{MSA}(\mathrm{LN}(x_l')) + x_l' \quad (6)$$

where LN represents the layernorm operation. From Equation 2 and Equation 4, we find that there are tons of matrix multiplication operations in the MSA and MLP modules. With such a big amount of matrix parameters, vision transformer usually comes with huge model volume. Therefore, we can effectively reduce the model size by quantizing the weights in each encoder's MSA and MLP modules, including $\mathbf{W}^q, \mathbf{W}^k, \mathbf{W}^v, \mathbf{W}^1, \mathbf{W}^2$. Furthermore, we also quantize activations in MLP. We do not quantize inputs and the attention maps in this paper since their data volume is not huge, and the quantization error may easily result in significant accuracy loss [8].

3.2. Quantizer for weights and activations

We apply a clipped uniform quantizer for both weights and activations in this paper, since the value distribution of weights/activations from the same layer usually concentrated in a smaller range. We set maximum and minimum values as clipping parameters to trim the data. The applied clipped uniform quantizer can be expressed as:

$$\mathrm{Q}(x, \alpha, b) = \frac{round((2^b-1)\cdot clip(x, -\alpha, \alpha))}{2^b-1} \quad (7)$$

where x is the weights or activations, b represents bit depth, the clipping threshold α is the maximum absolute value of x, $round(\cdot)$ is the round-off function. The uniform quantizer quantizes the input x to the range $[0, 2^b-1]$.

3.3. Transform

In this paper, we apply transform to decorrelate the weights, enabling efficient representation for each weight element. We propose a transform $\mathbf{K} = \mathbf{U^T W}$ on the weight matrix $\mathbf{W}$ and then allocate bits to the transformed weight matrix $\mathbf{K}$ to minimize the network output error. We determine the transform matrix by calculating the covariance matrix of weights and gradients. The transform matrix $\mathbf{U^T}$ is obtained by diagonalizing the matrix $\mathbf{C}_{\theta\theta}$, and $\mathbf{C}_{\gamma\gamma}^{-1}$,

$$\mathbf{C}_{\theta\theta} = \frac{1}{m}\sum_{j=1}^{m}\theta_j\theta_j^{\mathbf{T}}, \quad \mathbf{C}_{\gamma\gamma} = \sum_{j=1}^{m}\partial y/\partial\theta_j(\partial y/\partial\theta_j)^{\mathbf{T}} \quad (8)$$

where θ is the network hyper-parameters and y is the output of the network. $\mathbf{C}_{\theta\theta}$ and $\mathbf{C}_{\gamma\gamma}$ are the covariance matrix of the weights and gradients, respectively. To ensure the weight matrix $\mathbf{W}$ can be transformed to the identity matrix, the transform matrix $\mathbf{U^T}$ can be achieved by solving the following formulas:

$$\mathbf{U^T C}_{\theta\theta}\mathbf{U} = \Lambda, \quad \mathbf{U^T C}_{\gamma\gamma}^{-1}\mathbf{U} = \mathbf{I} \quad (9)$$

where Λ is a non-negative diagonal matrix, and $\mathbf{I}$ is the identity matrix.

In the inference process, for simplicity, we denote the transform matrix as $\mathbf{B} = \mathbf{U^{-T}}$. So that, the original weight matrix can be presented as:

$$\mathbf{W} = \mathbf{BK} \quad (10)$$

With the quantized $\mathbf{B}$ and $\mathbf{K}$:

$$\mathbf{B}^q = \mathrm{Q}(\mathbf{B}, -\alpha_B, \alpha_B), \mathbf{K}^q = \mathrm{Q}(\mathbf{K}, -\alpha_K, \alpha_K), \quad (11)$$

the forward process $y = x\cdot\mathbf{W^T} + \mathbf{b}$ of MSA and MLP module can be re-formulated as:

$$y = x\cdot(\mathbf{B}^q\mathbf{K}^q)^{\mathbf{T}} + \mathbf{b} = x\cdot(\mathbf{K}^q)^{\mathbf{T}}\cdot(\mathbf{B}^q)^{\mathbf{T}} + \mathbf{b} \quad (12)$$

3.4. Bit allocation

After the transform process, the transform matrices $\mathbf{B}$ and transformed weight $\mathbf{K}$ will be quantized with the scalar quantizer introduced in Sec.3.2. Assigning bits for each element is impractical, since it will result in very high computational complexity. Alternatively, we apply a block-wise bit allocation, where an entire matrix is divided into several small blocks, following [19]. It is worth noting that, for activations,

since transform is not applied, we can simply allocate bits by each layer. As such, in this paper, we allocate bits for weights in a block-wise manner and perform layer-wise bit allocation for activations to reduce time complexity.

Let $\widehat{y} = \sigma_q(f(x|\mathbf{B}^q\mathbf{K}^q))$ be the network output of the quantized model. The optimal bit allocation can be achieved by minimizing $\mathbb{E}\left\|\widehat{y} - y\right\|_2^2$, subject to given bit:

$$\begin{aligned} &\text{minimize MSE} = \mathbb{E}\left\|\widehat{y} - y\right\|_2^2 \\ &\text{subject to } R(b_1, ..., b_L) \end{aligned} \tag{13}$$

where $b_1, ..., b_L$ are the assigned bit depth numbers for block 1 to L. Then the optimal bit allocation is transformed into the problem of minimizing the mean square error between the $\widehat{y}$ and y. From [19], the above problem can be resolved as a Lagrangian conditional extremum problem, which can be fomulated as below:

$$\text{minimize } \mathcal{L} = \text{MSE} + \lambda R(b_1, ..., b_L) \tag{14}$$

$$\lambda = -\frac{\partial \text{MSE}}{\partial R} \tag{15}$$

where λ determines the balance between the bit allocation and the mean square error. And the optimal bit allocation can be obtained with Algorithm 1.

Algorithm 1: Optimal bit allocation

Input: $\mathbf{W}, \alpha, \lambda, MSE$
Output: b^{opt}

```
L = ∞, b_k = 0
for b = 1, ..., R do
    if L > MSE(b) + λb then
        L = MSE(b) + λb
        b^opt = b
    end
end
```

4. EXPERIMENTS

In this section, we first introduce the experimental implementation details. Then, the comparison results on ImageNet are presented. Thirdly, an ablation study is provided to check the effectiveness for each part of our proposed method. Finally, we discuss on the possibility to extend this method from post-training to re-training.

4.1. Implementation details

Datasets: ImageNet [20] dataset is employed to test the accuracy of vision transformers in image classification task. The training set contains more than 1.2 million nature images and the validation set contains 50,000 images with 1000 categories. In predicting phase, the transformer model will rank 1000 categories in order of probability from high to low. Top-1 accuracy is employed to assess the classification performance in the experiments.

Experimental settings: In our proposed model compression flow, the transform matrices and the bit allocation should be solved with image samples. In particular, we randomly select 50 images from the training set to obtain gradients of the weights for each transformer models to solve the transform matrices. We randomly select 100 images from the training set to calculate the optimal bit allocation.

4.2. Comparisons with state-of-the-art

We applied a variety of vision transformers on ImageNet to test the effectiveness of our method, including ViT[1], DeiT[21], and Swin Transformer[22]. Our proposed method is compared with four existing post-training quantization methods in the literature, including PTQforViT [8], FQ-ViT [10], PTQ4ViT [11], and PSAQ-ViT [9]. The experimental results are shown in Table 1. It is obvious that existing methods can only quantize vision transformers to 8-bit with accuracy drop less than 0.5%. On the contrary, our approach (TQ-ViT) can compress almost all transformer models (except DeiT-S) down to 6-bit with accuracy drop less than 0.5%. Compared to the existing methods, our proposed method obtains superior performance for all the transformer variants at all bit rate settings, only except $Swin - S@8bit$ where our method is slightly lower (0.02%) than PTQ4ViT. It can also be found that the advantage of our method become even greater at lower bit settings (i.e., 6-bit and 4-bit).

4.3. Ablation study

We conduct experiments with Swin-S on ImageNet to validate the effectiveness of two main components of our proposed method (i.e., transform and bit allocation). We evaluate the quantization performance with and without these two components, where Fig. 2 shows the corresponding experimental results. It can be observed that, without the proposed transform method, the cap accuracy we could achieve get decreased. It reflects the effectiveness of the transform which can decorrelate the weight matrices making scalar quantizer work better for each individual weight element. If further removing the bit allocation, the lost of optimization for network output errors makes the quantization performance drop dramatically. It can prove that both transform and bit allocation mechanisms are necessary for the proposed transform quantization method. The optimal results can be achieved only when both of these two modules are well integrated. In addition, model sizes of the full precision and the quantized TQ-ViT (on Swin-S) are shown in Table 2.

4.4. Discussion

When the network is quantized to extremely low-bit, such as 4-bit, the model accuracy drops a lot with post-training quantization. It makes us wonder whether our transform quantization method can be extended to quantization-aware training (QAT). We find that the answer could be yes.

The key of this method is to solve the transform matrices and bit allocation. Once the model weights and gradients are given, the proposed post-training method can guarantee the optimal transform and bit allocation parameters. We can

Table 1: Comparison of the top-1 accuracy with state-of-the-art methods on ImageNet dataset. TQ-ViT is our proposed method. 'MP' represents for mixed-precision.

Method	W/A	Swin-T	Swin-S	Swin-B	DeiT-T	DeiT-S	DeiT-B	ViT-B	ViT-L
Full Precision	32/32	81.38	83.23	83.60	72.13	79.83	81.80	84.54	85.83
PTQ for ViT[8]	8/8 MP	-	-	-	-	78.09	81.29	-	-
PTQ4ViT[11]	8/8	81.24	83.10	-	-	79.47	81.48	84.25	-
FQ-ViT[10]	8/8	80.51	82.71	82.97	71.61	79.17	81.20	83.31	85.03
PSAQ-ViT[9]	8/8	75.35	76.64	-	71.56	76.92	80.26	37.36	-
TQ-ViT(ours)	**8/8 MP**	**81.24**	83.08	**83.54**	**72.10**	**79.80**	**81.85**	**84.47**	**85.82**
PTQ for ViT [8]	6/6 MP	-	-	-	-	75.10	77.47	-	-
PTQ4ViT [11]	6/6	80.47	82.38	-	-	76.28	80.25	81.65	-
TQ-ViT(ours)	**6/6 MP**	**80.99**	**82.85**	**83.36**	**71.77**	**79.25**	**81.63**	**84.06**	**85.69**
PTQ for ViT	4/4 MP	-	-	-	-	-	75.94	-	-
TQ-ViT(ours)	**4/4** MP	**78.28**	**80.12**	**82.01**	**65.83**	**72.83**	**80.46**	**80.91**	**85.03**

Table 2: Model sizes of the full precision and the quantized TQ-ViT. 'MP' represents for mixed-precision.

Method	W/A	Model size(MB)	Comp ratio
Full Precision	32/32	190.87	-
TQ-ViT	8/8 MP	48.31	3.95x
	6/6 MP	36.60	5.21x
	4/4 MP	24.51	7.79x

Table 3: Initial results on extending the transform quantization method from post-training to re-training.

Method	W/A	DeiT-T	DeiT-S
Full Precision	32/32	72.13	79.83
TQ-ViT	4/4	65.85	72.83
TQ-ViT+re-training	**4/4 MP**	**71.90**	**78.12**

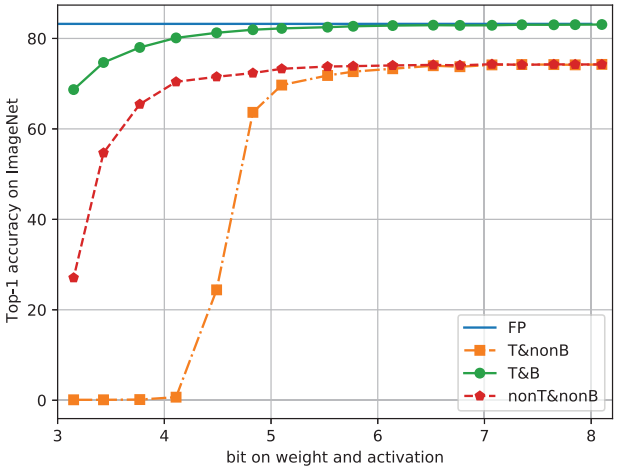

Fig. 2: Ablation study. 'FP' stands for Full Precision; 'T', 'B', 'nonT', and 'nonB' indicate w/ Transform, w/ Bit allocation, w/o Transform, and w/o Bit allocation respectively.

dynamically update the transform and bit allocation parameters during the re-training process, saying at the end of each training epoch. So that the quantization performance can keep optimal along with the re-training.

To validate this idea, we conduct a preliminary experiment. We re-train the quantized vision transform for a 5 epochs, and update the transform during the re-training at 4-bit, where Table 3 shows the results. Top-1 accuracy on quantized DeiT-T increases from 65.85% to 71.90%, and the accuracy on DeiT-S increases from 72.83% to 78.12%, which are remarkable. We believe quantization-aware training (QAT) for vision transformer with transform quantization could be an interesting topic to explore in future work.

5. CONCLUSION

In this paper, we proposed a transform quantization method for vision transformers. The method can solve the optimal transform matrices and bit allocation parameters to enable high-performance quantization in transformed domain. In particular, a proper transform was applied before quantization to decorrelate the weights, enabling efficient representation for each weight element. The quantization is then applied on the transformed weights with elaborated bit allocation, which is optimized to directly minimizing the network output errors. Fabulous post-training quantization performance was obtained over state-of-the-art methods. The proposed method can compress the transformer's weights and activations down to 6-bit without a significant task performance drop, whereas the state-of-the-art can only make 8-bit. Furthermore, we also explore the possibility to extend the transform quantization method from post-training to re-training, where initial results validated the feasibility. It could be an interesting work for future exploration.

6. REFERENCES

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al., "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko, "End-to-end object detection with transformers," in *ECCV*. Springer, 2020, pp. 213–229.

[3] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao, "Pre-trained image processing transformer," in *CVPR*, 2021, pp. 12299–12310.

[4] Yiming Hu, Siyang Sun, Jianquan Li, Xingang Wang, and Qingyi Gu, "A novel channel pruning method for deep neural network compression," *arXiv preprint arXiv:1805.11394*, 2018.

[5] Xuan Liu, Xiaoguang Wang, and Stan Matwin, "Improving the interpretability of deep neural networks with knowledge distillation," in *ICDMW*. IEEE, 2018, pp. 905–912.

[6] Emily L Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fergus, "Exploiting linear structure within convolutional networks for efficient evaluation," *NIPS*, vol. 27, 2014.

[7] Song Han, Huizi Mao, and William J Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *arXiv preprint arXiv:1510.00149*, 2015.

[8] Zhenhua Liu, Yunhe Wang, Kai Han, Wei Zhang, Siwei Ma, and Wen Gao, "Post-training quantization for vision transformer," *NIPS*, vol. 34, pp. 28092–28103, 2021.

[9] Zhikai Li, Liping Ma, Mengjuan Chen, Junrui Xiao, and Qingyi Gu, "Patch similarity aware data-free quantization for vision transformers," *arXiv preprint arXiv:2203.02250*, 2022.

[10] Yang Lin, Tianyu Zhang, Peiqin Sun, Zheng Li, and Shuchang Zhou, "Fq-vit: Post-training quantization for fully quantized vision transformer," in *IJCAI*, 2022, pp. 1173–1179.

[11] Zhihang Yuan, Chenhao Xue, Yiqi Chen, Qiang Wu, and Guangyu Sun, "Ptq4vit: Post-training quantization framework for vision transformers," *arXiv preprint arXiv:2111.12293*, 2021.

[12] Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L Yuille, and Yuyin Zhou, "Transunet: Transformers make strong encoders for medical image segmentation," *arXiv preprint arXiv:2102.04306*, 2021.

[13] Yuxin Fang, Bencheng Liao, Xinggang Wang, Jiemin Fang, Jiyang Qi, Rui Wu, Jianwei Niu, and Wenyu Liu, "You only look at one sequence: Rethinking transformer in vision through object detection," *NIPS*, vol. 34, pp. 26183–26197, 2021.

[14] Charles N Christensen, Meng Lu, Edward N Ward, Pietro Lio, and Clemens F Kaminski, "Spatio-temporal vision transformer for super-resolution microscopy," *arXiv preprint arXiv:2203.00030*, 2022.

[15] Pankaj Mishra, Riccardo Verk, Daniele Fornasier, Claudio Piciarelli, and Gian Luca Foresti, "Vt-adl: A vision transformer network for image anomaly detection and localization," in *ISIE*. IEEE, 2021, pp. 01–06.

[16] Dayang Wang, Zhan Wu, and Hengyong Yu, "Ted-net: Convolution-free t2t vision transformer-based encoder-decoder dilation network for low-dose ct denoising," in *IWMLMI*. Springer, 2021, pp. 416–425.

[17] Aojun Zhou, Anbang Yao, Yiwen Guo, Lin Xu, and Yurong Chen, "Incremental network quantization: Towards lossless cnns with low-precision weights," *arXiv preprint arXiv:1702.03044*, 2017.

[18] Dongqing Zhang, Jiaolong Yang, Dongqiangzi Ye, and Gang Hua, "Lq-nets: Learned quantization for highly accurate and compact deep neural networks," in *ECCV*, 2018, pp. 365–382.

[19] Wang Zhe, Jie Lin, Vijay Chandrasekhar, and Bernd Girod, "Optimizing the bit allocation for compression of weights and activations of deep neural networks," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 3826–3830.

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," *ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[21] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou, "Training data-efficient image transformers & distillation through attention," in *ICML*. PMLR, 2021, pp. 10347–10357.

[22] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10012–10022.