Utilizing the Attention Mechanism for Accuracy Prediction in Quantized Neural Networks

Lu Wei , Zhong Ma * , ChaoJie Yang , Qin Yao , Wei Zheng

Posted Date: 30 January 2025

doi: 10.20944/preprints202501.2272.v1

Keywords: neural network; quantization; accuracy; prediction; attention

Article

Utilizing the Attention Mechanism for Accuracy Prediction in Quantized Neural Networks

Lu Wei [1,2], Zhong Ma [2,*], ChaoJie Yang [2], Qin Yao [1,2] and Wei Zheng [1]

[1] School of Software, Northwestern Polytechnical University, Xi'an 710072, China

[2] Xi'an Microelectronics Technology Institute, Xi'an 710065, China

* Correspondence: mazhong@mail.com; Tel.: +86-187-1089-0519

Abstract: Quantization plays a crucial role in deploying neural network models on resource-limited hardware. However, current quantization methods have issues like large accuracy loss and poor generalization for complex tasks. These issues pose obstacles to the practical application of deep learning and large language models in smart systems. The main problem is our limited understanding of quantization's effect on accuracy, and there is also a need for more effective approaches to evaluate the performance of the quantized models. To address these concerns, we develop a novel method that leverages the self-attention mechanism. This method predicts a quantized model's accuracy using a single representative image from the test set. It utilizes the Transformer encoder and decoder to perform this prediction. The prediction error of the quantization accuracy on three types of neural network models is only 2.35%. The proposed method enables rapid performance assessment of the quantized models during the development stage, thereby facilitating the optimization of quantization parameters and promoting the practical application of neural network models.

Keywords: neural network; quantization; accuracy; prediction; attention

1. Introduction

Breakthroughs in artificial intelligence, driven by advanced convolutional neural networks and large language models, have made great progress in many fields, such as language comprehension, intelligent identification of satellites in orbit, and autonomous driving of vehicles [1,2]. However, these models with numerous parameters have high accuracy but are usually very large, which limits their practical use in resource-constrained environments. This has created an urgent need to accelerate the development of neural network models.

The quantization of neural network models is an important approach for optimizing models and increasing their speed [3]. It converts traditional 32-bit floating-point parameters into more compact 8-bit or lower precision formats [4,5]. This technique is crucial for the AI field because it can reduce storage, computational needs, and energy consumption of neural network models [6-8]. It enables complex neural networks to be effectively implemented on resource-limited platforms like mobile devices and embedded systems [9,10]. Quantization extends the application of deep learning and promotes the use of intelligent algorithms in edge computing environments [11,12]. It also provides a solid foundation for real-time data analysis, processing, and decision-making, enhancing the operational effectiveness of AI in industries like smart wearables, edge computing, and collaborative sensing [13,14].

Quantization involves converting continuous floating-point numbers to discrete integers, which brings benefits but also causes inherent precision loss [15-17]. Finding a quantization technique to minimize this loss is a major challenge in modern AI research. Although there are clear advantages, existing quantization methods often suffer from significant precision degradation and lack of

 doi:10.20944/preprints202501.2272.v1

robustness. This is mainly due to the unclear mechanisms that cause precision reduction during quantization, making it hard to accurately evaluate the performance of quantized models.

Quantization is a complex process affected by various factors, and each factor uniquely and interconnectedly influences the model's accuracy. Input datasets and quantization parameter choices can significantly impact quantization accuracy. Weight distributions branches vary greatly, resulting in a wider range of converted weights. A large part of values cluster around 0, but values away from 0 are also important. This combination makes quantification challenging [18]. Therefore, it is of great significance to select appropriate quantization parameters. Therefore, Mixed-Precision Quantization (MPQ) is a major trend in the quantization of neural network models in the future [19-21]. Different combinations of bit-widths will also lead to varying degrees of accuracy loss [22-24]. A common way to achieve quantization accuracy is to run the model on the entire test dataset. But this method has limitations, such as scalability problems and the risk of overfitting to the test data, which may not work well with new, unseen data. Moreover, running the model on a full test set can be computationally costly and time-consuming, especially for large models and datasets.

From the dataset perspective, large test sets require significant storage and processing time, while small ones may give unreliable results. Research shows that compressing datasets can reduce data volume and improve model validation efficiency. Dataset Distillation was first proposed by Wang et al. [25], aiming to extract knowledge from large datasets to create much smaller synthetic datasets. Models trained on these distilled datasets are expected to perform as well as those trained on the original larger datasets. Zhao et al. [26] proposed a dataset compression technique called Dataset Condensation. It aims to train deep neural networks effectively by generating a small number of information-rich synthetic samples, reducing storage and processing costs of large datasets. Sajedi et al. [27] introduced a new dataset distillation method, DataDAM, which uses the Spatial Attention Matching Module (SAM) to capture and replicate the original dataset's distribution characteristics. Guo et al. [28] put forward Difficulty-Aligned Trajectory Matching (DATM), which generates targeted informative patterns based on synthetic data's learning trajectories during training. It focuses on reproducing basic easy-to-learn patterns for small synthetic datasets and emphasizes modeling more complex patterns for larger ones. However, existing dataset compression techniques mainly focus on compressing training datasets, often ignoring the test dataset. As a result, the test dataset, which is important for evaluating model performance, may remain large and cumbersome, reducing the overall efficiency gains of dataset compression.

From the hardware perspective, when testing with simulated quantization on a PC, fully simulating the hardware computation process is very time-consuming. Simplifying the quantization computation process, as in fake quantization [29], can lead to differences between simulated results and actual hardware performance due to different implementation methods. If testing is done on actual end devices, all samples must be evaluated to obtain performance metrics. This testing method has a limited application scope and requires more time and storage space. Processing and reasoning through all test dataset samples take a lot of time and storage, making it impractical in deployment scenarios with limited computing resources.

From the model structure perspective, existing methods for predicting the accuracy of quantized neural networks are usually limited to specific architectures. They cannot adapt well to changes in model structures and fail to effectively encode the representation of quantization parameters. Wang et al [30] constructed a quantization accuracy predictor based on the highly flexible Once For All network [31]. This predictor encodes the model structure and quantization strategy to directly predict the accuracy of the quantization model. However, in this approach, collecting the quantization dataset requires 16,000 GPU hours, which is both costly and time-consuming. Moreover, the quantization predictor is only capable of predicting neural network models with a predetermined structure.

To address these challenges, we propose a single-image input accuracy predictor for neural network models using the attention mechanism. This predictor uses an encoder to learn the features of each layer of the quantized data and then generates a single image representing the whole dataset

 doi:10.20944/preprints202501.2272.v1

through a decoder. By taking a single image as input, it can quickly output the model's quantization accuracy, providing a more efficient and practical solution for evaluating quantized models.

2. Methods

2.1. Overall Design Scheme

Existing methods for measuring the accuracy loss of the quantized neural network models usually require a series of steps on the embedded side. Besides, the quantization accuracy loss is measured by comparing the differences in the model's output before and after quantization. However, these methods are time-consuming and complex when deploying models on the embedded side. This is not beneficial for the instant iterative optimization of the quantization parameters and the rapid development of the tool chain.

We conduct research on methods for measuring the accuracy loss of the quantized neural network models, aiming to avoid actual measurements on the embedded side and predict the computational accuracy of models with different quantization parameters. We construct an accuracy predictor based on the self-attention mechanism to evaluate the quantization accuracy loss of neural network models with different bit widths. First, we use the Transformer structure to encode the feature maps of each layer in the quantized neural network and apply the attention mechanism to understand how much each layer of the neural network impacts the accuracy after quantization. Based on this, we implement the encoder and decoder based on the Transformer structure to generate single - image data instead of using the whole test set, thus significantly reducing the testing time.

The benefits of using the attention mechanism to evaluate the accuracy of quantized neural networks include:

Learning Relevance: By learning the importance of each layer of the neural network model after quantization, the self-attention mechanism can capture the intricate dependencies within the data, and automatically analyze and learn the factors affecting the accuracy of the quantized model.

Capturing Diverse Information: Each head can focus on different parts of the quantized data by attending to different parts of the sequence in parallel. For example, one head may focus on the rounding error caused by quantization, while another may focus on the quantization clipping error.

Enhancing Representational Capacity: Processing multiple aspects of the input simultaneously enables the model to form a more comprehensive understanding of the quantized data, resulting in better performance on complex tasks.

Improving Prediction Efficiency: It allows for parallel computation as all layers can be processed simultaneously, unlike Recurrent Neural Networks (RNNs) and Long Short - Term Memory (LSTM) networks that process elements sequentially.

The accuracy evaluator we developed, as shown in Fig. 1, consists of two parts: an accuracy predictor based on the multi-head self-attention mechanism and a data generator based on the Transformer encoder and decoder. The entire training process is divided into two stages. The first stage is the training convergence of the accuracy predictor. For the trained mixed bit width quantization model and its test dataset, the output feature maps of each layer of the quantized model are processed with a histogram and then input into the accuracy predictor for training. The convergence condition is that the accuracy predictor outputs the prediction accuracy of the quantized model within the normal range. The second stage is the training convergence of the encoder - decoder and the data generator. Fixing the accuracy predictor from the first stage, we train the decoder and data generator of the Transformer structure. Then, we input the single-image generated by the data generator into the quantization model and generate the prediction accuracy through the output of the accuracy predictor.

 doi:10.20944/preprints202501.2272.v1

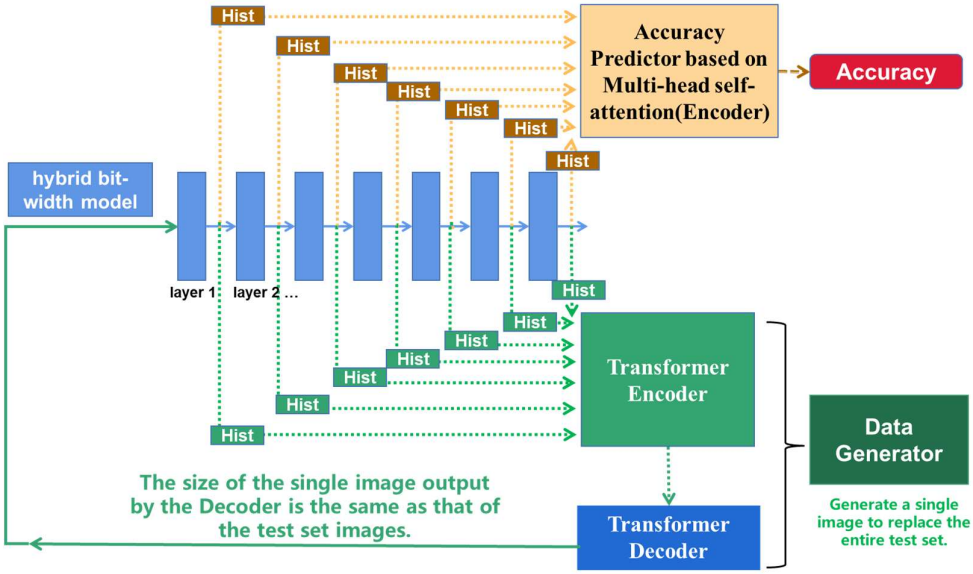

Figure 1. Attention mechanism based accuracy predictor using a single representative image from the test set.

2.2. Accuracy Predictor based on Multi-head self-attention

As depicted in the upper part of Fig. 1, the accuracy predictor, based on multi-head self-attention, takes a trained mixed bit-width quantized model as input. Here, both the quantization parameters and the quantization strategy can vary. The role of the accuracy evaluator is to forecast the accuracy of this input model. Initially, we must acquire the output feature maps of each layer of the input model. Given that the dimensionality of the feature maps for each layer differs, they are processed into a histogram matrix of a fixed dimensional size. We generate the histograms for the data from each layer of the quantized neural network model. In one branch of the histogram, the height represents the frequency of the data. Given the varying amounts of data in each layer of the neural network model, the other branch normalizes the histogram by converting the height to a rate. Subsequently, the two types of histograms are concatenated using the Concat operator and utilized as the input sequence to the Transformer Encoder, as shown in Fig. 2. Furthermore, the input sequence needs to be added with position encoding [32]. For each position in the token embedding, the positional encoding is added as shown in (1).

$$PE(t) = \begin{cases} \sin\left(\frac{pos}{10000^{\frac{t}{d}}}\right), & if\ t = 2i \\ \cos\left(\frac{pos}{10000^{\frac{t}{d}}}\right), & if\ t = 2i + 1 \end{cases} \quad (1)$$

where pos is the index of time step in the token embedding and d is the vector dimension.

 doi:10.20944/preprints202501.2272.v1

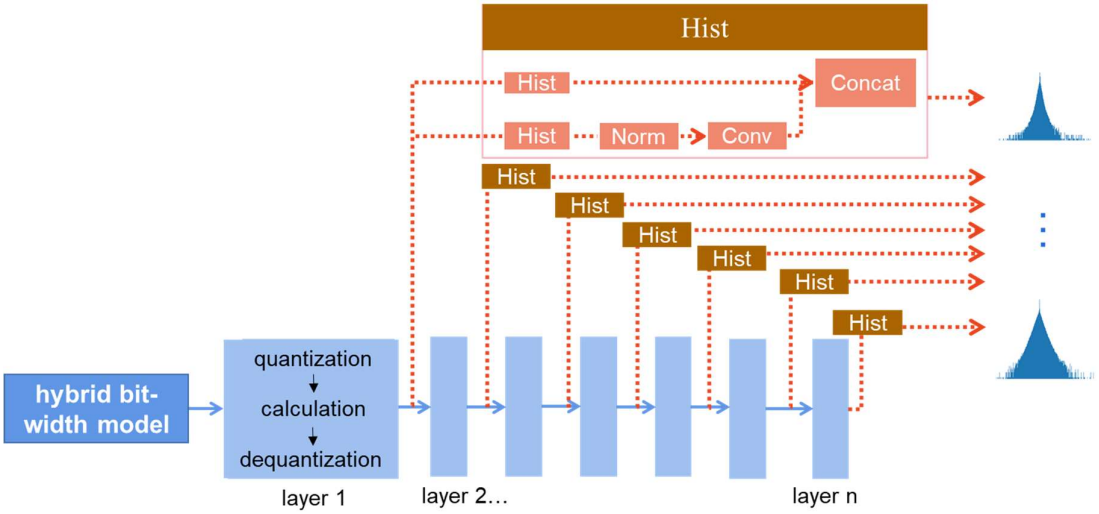

Figure 2. The process of obtaining the input sequence of the Transformer Encoder from the quantized data.

For each layer or operator of a neural network model, the processes of quantization, calculation, and dequantization are required. Quantization refers to the process of converting data from a floating-point type to an integer type, as indicated in (2). Dequantization, on the other hand, is the process of mapping the computation results of the integer type back to the corresponding floating-point type, as shown in (3). This is the method through which the floating-point results after quantization are obtained.

$$Q = quantization(\mathrm{r}) = round\left(\frac{clamp(r,\alpha,\beta) - D}{\mathrm{s}}\right) \tag{2}$$

$$r' = dequantization(Q) = s \cdot Q + D \tag{3}$$

where r is the original floating-point value, Q is the integer value obtained after quantization, s and D are quantization parameters. Here, s represents the scaling factor, and D represents the zero point, which is selected in such a way that the value of 0 can be precisely mapped to quantized values. $[\alpha, \beta]$ is the clipping threshold for all the floating-point data of one layer. The clipping thresholds have a significant influence on the computation of quantization parameters. The clamp function is a mathematical function that constrains a given value within a specified range, which is referred to as the minimum bound and maximum bound. The round function is the fundamental function used to round numbers to the nearest integer.

Different quantization parameters will lead to different precision loss. We can analyze the influence of different quantization parameters on data distribution by the histograms. The processed histograms are then fed into the accuracy evaluator, and through the multi-head self-attention mechanism, autonomous training and learning commence. Using accuracy as the loss label, the accuracy evaluator is trained to assign coefficients to the output of each layer. It also learns the correlations among different feature maps in the input, thus completing the direct modeling from the mixed bit-width quantization model to task accuracy.

In the proposed method, the multi-head self-attention adopts the ViT (Vision Transformer) model [33]. The concept of ViT involves segmenting an image into small patches. These patches are then used as a linear embedding and input into the Transformer, processed in a manner similar to tokens in NLP, with supervised training for image classification. The self-attention and multi-head self-attention (MHSA) [32,34] is shown in (4-6).We substitute the classification token with a token that outputs the prediction accuracy. Employing an attention-based mechanism to learn the feature maps of the quantization model's layers can enhance the ability to extract local features, to the extent that it reveals how different layers impact the quantization accuracy.

 doi:10.20944/preprints202501.2272.v1

$$Self - attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{D_K}})V \quad (4)$$

$$head_i = Self - attention(Q_i, K_i, V_i), i = 1,2, \dots n \quad (5)$$

$$MHSA(Q, K, V) = [head_1; head_2; \dots head_n] * W \quad (6)$$

where Q, K and V represent Query (Q), Key (K), and Value (V) matrices of the self-attention mechanism respectively. The outputs from all heads $\{head_1,\ head_2, \dots\ head_i\}$ are concatenated and passed through a single FC layer of dimensions W.

2.3. Data generator based on Transformer Encoder and Decoder

The data generator, depicted at the bottom of Fig. 1, is composed of an encoder and a decoder. It fuses the output feature maps of all the layers of the mixed bit-widths quantized model and inputs them into the Transformer-based encoder. The output of the encoder is used as the input of the decoder. The decoder outputs a single image that can represent the entire test set.

The encoder uses the self-attention mechanism to learn the correlation between the feature maps of the layers, which is in line with the principle of the accuracy predictor in the upper part of Fig. 1. It takes the fused feature maps from the data generator as input and processes them to generate an output that will be fed into the decoder.

The decoder component of the data generator consists of a stack of identical layers. Each layer contains a multi-head self-attention mechanism and a position-wise fully connected feed-forward network. Additionally, it features a masked self-attention module, which ensures that the predictions for a position can only depend on known outputs at positions preceding it. This helps maintain the auto-regressive nature of the sequence generation process. The Transformer decoder was first introduced by Ashish Vaswani et al [32] and has since been widely used in computer vision tasks [35-39]. Our work that utilizes the Transformer decoder is inspired by the Masked Generative Image Transformer [40]. The decoder of MaskGIT [40] is designed based on a bidirectional Transformer architecture, incorporating various key operators to achieve efficient image generation functions. Its structure is built around a bidirectional self-attention mechanism, which can fully capture the global dependency relationships between image tokens and provide abundant contextual information for accurate token prediction. By integrating operators such as positional embedding and layer normalization, the model's performance is further optimized to ensure stable image generation during both training and inference processes. The decoder yields a solitary image capable of representing the entirety of the test set, so that the accuracy predictor can use the single representative image as the input for accuracy prediction in practical use, reducing the computational cost of the accuracy predictor, and improving the iterative efficiency of the development of neural network models.

2.4. The process of training and using the accuracy predictor

How to train and use the accuracy evaluator is shown in Fig. 3. The training is divided into two stages. In stage 1, multiple images are used as input and the Transformer Encoder is trained so that it can learn how much each layer of the neural network model affects the accuracy and get the predicted value of the accuracy. In stage 2, multiple images are used as input, the Transformer Encoder is fixed, and the Transformer Decoder is trained so that it can learn the features of the dataset and generate the single test image. After the training is completed, the accuracy predictor is used in such a way that the single test image is used as input, and the accuracy prediction value of the complete dataset is obtained after the transformer encoder.

 doi:10.20944/preprints202501.2272.v1

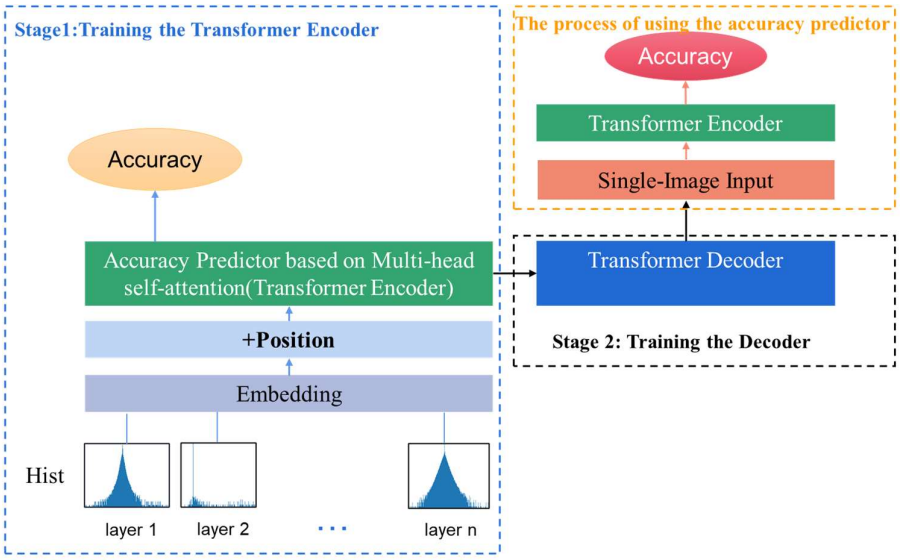

Figure 3. The process of training and using the accuracy predictor.

3. Implementation and Experimental Results

3.1. Experimental Setting

In this research, we employ convolutional neural networks (CNNs), which have been trained on established datasets, to address the well-known problem of image classification. The CNN architectures selected for our experiments are MobileNetV2, ResNet50, and VGG16. To investigate the impact of quantization, the neural networks used in our experiments have been quantized to different bit-widths, namely INT8, INT4, and INT2, and also in mixed formats. It is crucial to note that each unique quantization parameter leads to a different level of accuracy, which forms a fundamental aspect of our study.

The core objective of our experimental design is to demonstrate the effectiveness of a newly developed quantization accuracy predictor. In the domain of image classification, we have adopted the Top-1 Accuracy as our evaluation metric. This metric calculates the proportion of cases where the model's top prediction matches the actual class, thus offering a straightforward and accurate measure of the model's performance.

Our validation is carried out on a computer system that features an Intel(R) Core(TM) i7-8700K processor running at 3.70GHz, complemented by an NVIDIA GeForce GTX1070 graphics card. The proposed accuracy predictor is used to evaluate the performance of all quantized models. Subsequently, the predicted results are compared with the actual accuracy to determine the prediction error. It should be emphasized that the actual accuracy refers to the empirical label accuracy obtained through testing.

To our knowledge, this paper provides the first evidence demonstrating the efficacy of modeling for quantization accuracy prediction on image classification. Since the current evaluation of the accuracy of quantized models either requires testing the accuracy on a complete test set, which is time-consuming, or can only perform accuracy prediction on neural network models with specific structures, thus having application limitations. The method we provide uses a single image instead of the entire test set, and the accuracy of the quantized model can be evaluated without actual testing. Currently, there is a lack of similar methods. Therefore, we are unable to directly compare our predictor with existing quantization evaluation methods. Instead, we benchmark our predictor's accuracy against the empirical labels. This strategy allows us to assess the practical viability of our approach.

 doi:10.20944/preprints202501.2272.v1

3.2. Dataset

The results presented in this section are derived from two datasets. The MobileNetV2 is evaluated using the UC Merced Land Use Dataset [41]. The UC Merced Land Use Dataset is widely recognized in the fields of computer vision and machine learning, particularly for tasks associated with land cover classification and urban mapping. It was developed by the University of California, Merced, and consists of a set of high-resolution aerial images that showcase various land use patterns in the city of Merced, California. This dataset stands out due to its high spatial resolution, approximately 0.5 meters per pixel, which enables the identification of small and detailed features typical of different land use types. The images are large, typically measuring 1000 x 1000 pixels, providing abundant visual information for analysis. It encompasses 21 distinct land use categories, ranging from urban features like commercial and residential areas to natural landscapes such as grasslands, trees, and water bodies. The complexity and variability of urban landscapes within this dataset make it a challenging yet valuable resource for testing the robustness and accuracy of algorithms.

For ResNet50 and VGG16, we utilize the CIFAR-10 dataset [42]. The CIFAR-10 dataset comprises 60,000 32x32 color images, divided into 10 different classes, with 6,000 images per class. This dataset is a staple in machine learning research for image classification tasks and serves as a benchmark for evaluating various algorithms' performance. CIFAR-10 poses a challenge due to the variety and complexity of its images. Given their small size, the task of image classification becomes more difficult compared to using larger images. Additionally, there is significant intra-class variation, meaning that objects within the same class can have diverse appearances. Researchers and practitioners frequently use CIFAR-10 to experiment with different image processing techniques, feature extraction methods, and machine learning models. It has been instrumental in the development and testing of convolutional neural networks (CNNs), which have proven highly effective for image classification tasks.

3.3. Main Results

Using a single image as input, our trained accuracy predictor is used to measure the precision of several models with different quantization parameters. For MobileNetV2, we observe an average prediction error of 2.77%, with detailed results presented in Table 1 and Fig. 4. For ResNet50, the average prediction error is 2.40%, as shown in Table 2 and Fig. 5. For VGG16, the average prediction error is 1.89% presented in Table 3 and Fig. 6. The reason why VGG16 performs better is that its model structure is a straightforward type, which is more friendly to quantization. Our findings emphasize the efficacy of our proposed method in predicting the quantization accuracy of neural network models within classification tasks. The proposed predictor is highly capable of quickly assessing the influence of input data, quantization parameters, bit-widths, and model architecture on the accuracy of neural network models. It exhibits versatility, being able to handle neural networks with diverse architectures and various quantization schemes, thereby enhancing its value in the field of machine learning.

Table 1. Accuracy prediction results of quantized MobileNetV2 models using a single representative image from the test set.

Model Index	label Accuracy	Prediction Accuracy	Prediction Error |Prediction Accuracy-label Accuracy|
1	84.31%	81.49%	2.82%
2	82.39%	79.38%	3.01%
3	85.55%	82.27%	3.28%
4	82.42%	80.11%	2.31%
5	78.40%	75.99%	2.41%

 doi:10.20944/preprints202501.2272.v1

Average	-	-	2.77%

Table 2. Accuracy prediction results of quantized ResNet50 models using a single representative image from the test set.

Model Index	label Accuracy	Prediction Accuracy	Prediction Error |Prediction Accuracy-label Accuracy|
1	88.58%	86.75%	1.83%
2	87.21%	84.59%	2.62%
3	86.38%	84.11%	2.27%
4	83.31%	81.17%	2.14%
5	85.13%	82.01%	3.12%
Average	-	-	2.40%

Table 3. Accuracy prediction results of quantized VGG16 models using a single representative image from the test set.

Model Index	label Accuracy	Prediction Accuracy	Prediction Error |Prediction Accuracy-label Accuracy|
1	87.55%	85.32%	2.23%
2	87.55%	86.04%	1.51%
3	85.45%	83.12%	2.33%
4	87.16%	86.03%	1.13%
5	85.45%	83.21%	2.24%
Average	-	-	1.89%

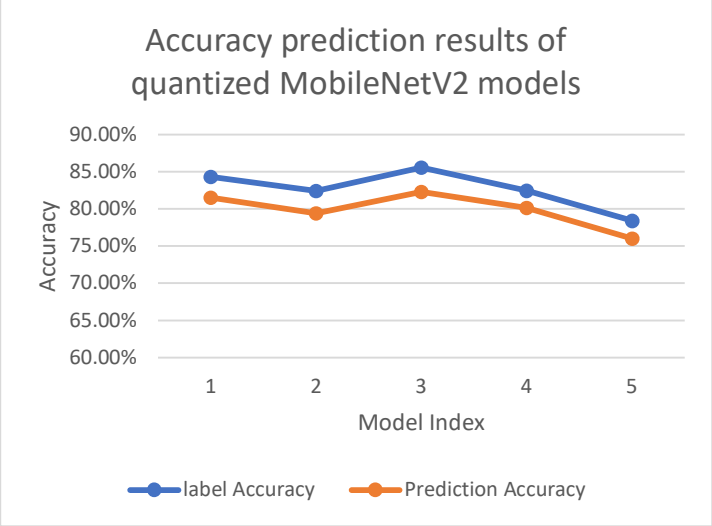

Figure 4. Accuracy prediction results of quantized MobileNetV2 models.

 doi:10.20944/preprints202501.2272.v1

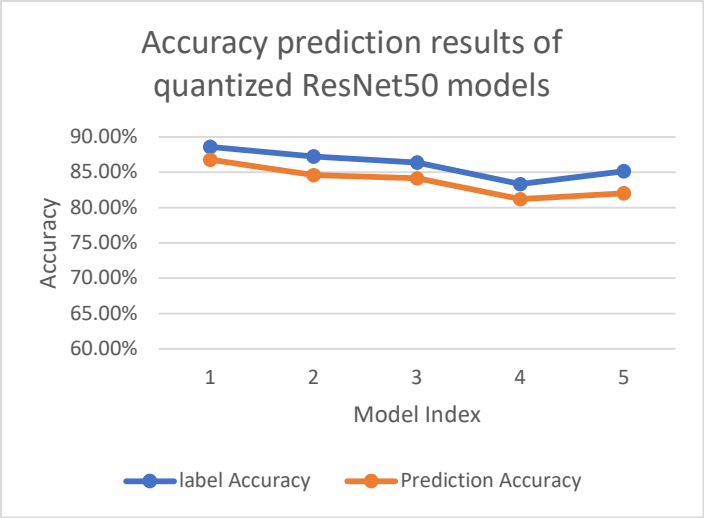

Figure 5. Accuracy prediction results of quantized ResNet50 models.

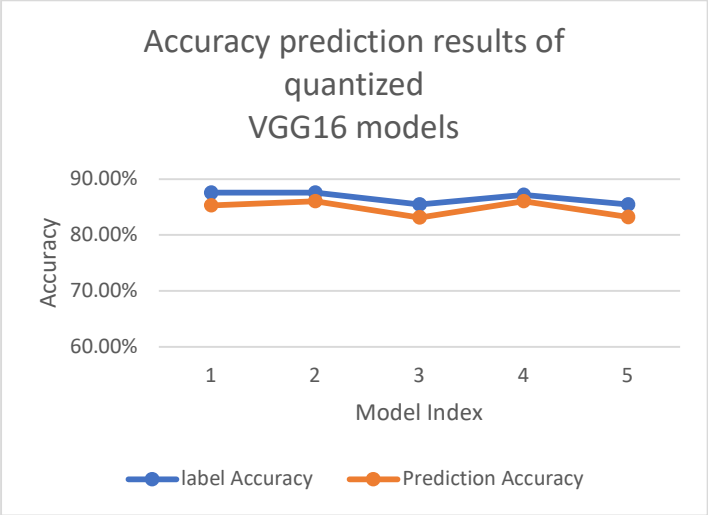

Figure 6. Accuracy prediction results of quantized VGG16 models.

3.4 Ablation Study

As depicted in Fig. 1, the accuracy evaluator for the quantized neural networks we developed is composed of two parts. One is an accuracy predictor founded on the multi-head self-attention mechanism, and the other is a data generator based on the Transformer encoder and decoder. The accuracy predictor in the first part can predict the accuracy of the neural network model after quantization. The data generator in the second part can generate a single image that represents the entire test set. Using this image as the input, the accuracy of the quantized model can be evaluated by the predictor in the first part, which significantly shortens the prediction time. We conduct the ablation study regarding whether using the single representative image. The results of the ablation experiment are shown in Table 4.

 doi:10.20944/preprints202501.2272.v1

Table 4. Ablation study on the effects of the single representative image. "Part 1" denotes the scenario where only the accuracy predictor in the first part is utilized, and the test set employed remains the complete original test set. "Part 1 + Part 2" indicates the situation in which a single image, capable of replacing the entire test set, is used for testing.

Model	Average Prediction Error (Part 1)	Average Prediction Accuracy (Part 1 + Part 2)
MobileNetV2	2.05%	2.77%
ResNet50	1.91%	2.40%
VGG16	1.41%	1.89%

The experimental results turned out as expected. In the "Part 1 + Part 2" approach, a single image generated by the data generator serves as a substitute for the entire test set. This significantly reduces the time needed for accuracy prediction, though at the expense of a minor drop in prediction accuracy. When compared with "Part 1", "Part 1 + Part 2" shows an average accuracy prediction loss of 0.56% across three different types of neural network models. If the priority is to enhance the speed of the predictor, the comprehensive "Part 1 + Part 2" method proposed in this paper is a viable option. On the other hand, if the focus is on maximizing the accuracy of the accuracy predictor, it is advisable to solely adopt the method presented in "Part 1".

4. Conclusions

In this paper, we conduct a study on the modeling and prediction of accuracy loss for quantized neural network models. We propose an innovative accuracy predictor that is founded on the attention mechanism. To enhance the computational efficiency of this predictor, a single image representing the entire test dataset is generated and utilized as input, leveraging the Transformer architecture. Experiments prove that the proposed method can effectively predict the accuracy of quantized neural network models of classification tasks. This method facilitates the rapid and convenient compression of models, which is of vital importance for deploying deep neural networks and large language models on resource-constrained devices.

Author Contributions: Conceptualization, L.W. and Z.M.; methodology, L.W.; software, C.Y.; vali dation, C.Y. and Q.Y.; formal analysis, L.W. and Z.M.; investigation, L.W. and W.Z.; resources, L.W. and W.Z.; data curation, C.Y.; writing—original draft preparation, L.W.; writing—review and editing, Z.M.; project administration, Z.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Datasets are available at UC Merced Land Use Dataset and CIFAR-10 and CIFAR-100 datasets.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhou X; Wu W. Unmanned system swarm intelligence and its research progresses. Microelectronics & Computer, vol. 38, no. 12, 2021, pp. 1-7. DOI: 10.19304/J.ISSN1000-7180.2021.1171.
2. Tang L; Ma Z; Li S; Wang Z. The present situation and developing trends of space-based intelligent computing technology. Microelectronics & Computer, vol. 39, no. 4, 2022, pp. 1-8. DOI: 10.19304/J.ISSN1000-7180.2021.1229.
3. Protsenko V; Kryzhanovskiy V; Filippov A. Quantization-Friendly Winograd Transformations for Convolutional Neural Networks. European Conference on Computer Vision.Springer, Cham, 2025, DOI:10.1007/978-3-031-73636-0_11.
4. Zhou S; Wu Y; Ni Z; Zhou X; Wen H; Zou Y. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. arXiv preprint arXiv:1606.06160, 2016.
5. Nagel M.; Fournarakis M.; Amjad R. A.; Bondarenko Y.; Baalen M. V.; Blankevoort T.. A White Paper on Neural Network Quantization. arXiv preprint arXiv:2106.08295, 2021.

6. Gholami A.; Kim S.; Dong Z.; Yao Z.; Mahoney M. W.; Keutzer K.. A Survey of Quantization Methods for Efficient Neural Network Inference. arXiv preprint arXiv.2103.13630, 2021.
7. Agustsson E; Mentzer F; Tschannen M; et al. Soft-to-hard vector quantization for end-to-end learning compressible representations. arXiv preprint arXiv:1704.00648, 2017.
8. Agustsson E; Theis L. Universally Quantized Neural Compression. 2020. DOI:10.48550/arXiv.2006.09952.
9. Kosunalp S. FPGA-QNN: Quantized Neural Network Hardware Acceleration on FPGAs. Applied Sciences, 2025, 15. DOI:10.3390/app15020688.
10. Zhang Y; Wang R; Zhang Y; et al. Mixed precision quantization of silicon optical neural network chip. Optics Communications, 2025, 574. DOI:10.1016/j.optcom.2024.131231.
11. Pei S; Wang J; Chen Y M. DPQ: dynamic pseudo-mean mixed-precision quantization for pruned neural network. Machine learning, 2024, 113(7):4099-4112. DOI:10.1007/s10994-023-06453-3.
12. Diao H; Hao Y; Xu S; et al. Implementation of Lightweight Convolutional Neural Networks via Layer-Wise Differentiable Compression. Multidisciplinary Digital Publishing Institute, 2021(10). DOI:10.3390/S21103464.
13. Hubara I; Nahshan Y; Hanani Y; et al. Accurate post training quantization with small calibration sets. International Conference on Machine Learning, PMLR, 2021, pp. 4466–4475.
14. Cai Y; Yao Z; Dong Z; et al. A novel zero shot quantization framework. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 13169–13178.
15. Osamaa A; Gadallaha S.; et al. Chaotic neural network quantization and its robustness against adversarial attacks. Knowledge-BasedSystems, 286, 2024.
16. Liang T; Glossner J; Wang L; et al. Pruning and quantization for deep neural network acceleration: A survey. arXiv preprint arXiv:2101.09671, 2021.
17. Wang Y; Liu Q. AQA: An Adaptive Post-Training Quantization Method for Activations of CNNs. IEEE Transactions on Computers, 2024. DOI:10.1109/TC.2024.3398503.
18. Yang D; He N; Hu X; Yuan Z; et al. Post-training Quantization for Re-parameterization via Coarse & Fine Weight Splitting. Journal of Systems Architecture, 147, 2024.
19. Dong P; Li L; Wei Z; et al. Emq: Evolving training-free proxies for automated mixed precision quantization. Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023: 17076-17086.
20. Tang C; Ouyang K; Chai Z; et al. SEAM: Searching Transferable Mixed-Precision Quantization Policy through Large Margin Regularization. Proceedings of the 31st ACM International Conference on Multimedia, 2023, 7971-7980.
21. Dong Z; Yao Z; Cai Y; Arfeen D; Gholami A; Mahoney M.W.; Keutzer K. Hawq-v2: Hessian aware trace-weighted quantization of neural networks. Advances in neural information processing systems, 2020.
22. Tang C; Ouyang K; Wang Z; et al. Mixed-Precision Neural Network Quantization via Learned Layer-wise Importance. 2022. DOI:10.48550/arXiv.2203.08368.
23. Choukroun Y; Kravchik E; Kisilev P. Low-bit quantization of neural networks for efficient inference. IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), IEEE, 2019, pp. 3009–3018.
24. Wang P; Chen Q; He X; et al. Towards accurate post-training network quantization via bit-split and stitching, in: International Conference on Machine Learning, PMLR, 2020, pp. 9847–9856.
25. Wang T.; Zhu J. Y.; Torralba A.; Efros, A. A.. Dataset distillation. arXiv preprint arXiv:1811.10959, 2018.
26. Zhao B.; Mopuri K. R.; Bilen H. Dataset condensation with gradient matching. In International Conference on Learning Representations, 2021.
27. Sajedi A.; Khaki S.; Amjadian E.; Liu L. Z.; Lawryshyn Y. A.; Plataniotis K. N.. Datadam: Efficient dataset distillation with attention matching. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 17097-17107.
28. Guo Z.; Wang K.; Cazenavette G.; Li Hui, Zhang Kaipeng; You Yang. Towards lossless dataset distillation via difficulty-aligned trajectory matching. In The Twelfth International Conference on Learning Representations, 2023.
29. Jacob B., Kligys S.; Chen B.; Zhu M.; Tang M.; Howard A; Adam H; Kalenichenko D. Quantization and training of neural networks for efficient integer-arithmetic-only inference. arXiv preprint arXiv:1712.05877, 2017.
30. Wang T; Wang K; Cai H; et al. APQ: Joint Search for Network Architecture, Pruning and Quantization Policy. in Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2006.08509, pp. 2075-2084, 2020.
31. Wang P; Yang A; Men R; et al. Unifying Architectures, Tasks, and Modalities Through a Simple Sequence-to-Sequence Learning Framework. in Proc. International Conference on Machine Learning, pp. 23318-23340, 2022.
32. Vaswani A.; Shazeer N.; Parmar N.; Uszkoreit J.; Jones L.; Gomez A. N, Kaiser L. Attention Is All You Need. arXiv, 2017. DOI: 10.48550/arXiv.1706.03762.
33. Dosovitskiy A.; Beyer L.; Kolesnikov A.; Weissenborn D.; Zhai Xiaohua. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In International Conference on Learning Representations, 2021.

 doi:10.20944/preprints202501.2272.v1

34. Krishna T; Murali E; Venkatram V; Arun K. Neural Architecture Search for Transformers: A Survey. IEEE Access, 2022, 10, pp. 108374-108412. DOI:10.1109/ACCESS.2022.3212767
35. Liu, Y.; Zhang, Y.; Wang, Y.; Hou, F.; et al. A survey of transformers in computer vision. arXiv preprint arXiv:2111.06091, 2021.
36. Carion Nicolas; et al. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision (ECCV), 2020.
37. Yuan L.; Chen Y.; Wang T.; Yu W.; Shi Y.; Tay F. E; et al. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021.
38. Liu Z.; Lin Y.; Cao Y.; Hu H.; Wei Y.; Zhang Z.; et al. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021.
39. Wang W.; Xie E.; Li X.; Fan D. P.; Song K.; Liang D.; et al. PVT v2: Improved baselines with Pyramid Vision Transformer. Computational Visual Media, 2022. DOI: 10.1007/s41095-021-0229-5.
40. Chang H; Zhang H; Jiang L; et al. MaskGIT: Masked Generative Image Transformer. 2022.DOI:10.48550/arXiv.2202.04200.
41. Yang Y; Newsam S. Bag-Of-Visual-Words and Spatial Extensions for Land-Use Classification. ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS), 2010.
42. Krizhevsky A; Hinton G. Learning multiple layers of features from tiny images. Handbook of Systemic Autoimmune Diseases, 2009, 1(4).