Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 10, No. 2, 2024

Improving the Performance and Efficiency of Convolutional Neural Networks (CNNs) on Image Classification Tasks via Fixed-Point Quantization and Structured Pruning

Yingjie Song *

Lappeenranta University of Technology, Lahti, 53850, Finland
*** Corresponding author Email:** Yingjie.Song@student.lut.fi

Abstract: This study explores the combined use of fixed-point quantization and structured pruning to optimize the performance and efficiency of convolutional neural networks (CNNs) in image classification tasks. These techniques can be used to reduce model size and computational complexity, making CNNs more suitable for deployment in resource-constrained environments such as mobile devices and embedded systems. Fixed-point quantization methods can reduce the bit-width of weights and activations, thereby reducing the computational load and memory footprint. On the other hand, structured pruning systematically removes unimportant convolutional filters or channels, which further reduces the model size and increases the inference speed. An experimental evaluation was performed on the ImageNet dataset using the ResNet-50 architecture. The results show that the combined strategy of quantization and pruning reduces the model size by up to 75% and increases the inference speed by 50%, while maintaining a classification accuracy of 74.5%, compared to 76.4% for the baseline model. Considering the significant increase in efficiency, a slight decrease in accuracy is acceptable. The results show that the integrated approach effectively compresses and accelerates the CNN model without a significant drop in accuracy, making it ideal for real-time applications.

Keywords: Convolutional Neural Networks; Fixed-point Quantization; Structured Pruning; Model Compression; Image Classification.

1. Introduction

Convolutional Neural Networks (CNNs) have a wide range of application scenarios in the field of computer vision. For example, image classification, target detection, face recognition, image generation, etc. CNNs can extract multi-layered features of an image through layers of stacked convolutional kernels, thus realizing highly complex vision tasks. Although CNNs are excellent in performance, the high demand of computational and storage resources has become an urgent problem in applications. Facing this challenge, how to compress the model parameter size and reduce the computational complexity without significantly degrading the model performance has become an important research direction. To solve this problem, researchers have proposed a variety of model optimization techniques, among which quantization and pruning are the two most used methods. The objective of this study is to combine fixed-point quantization and structured pruning techniques to design a CNN optimization strategy that can operate efficiently in resource-constrained environments.

2. Quantitative Techniques

Fixed-point quantization converts the floating-point weights and activation values of a model from 32-bit precision to 8-bit or even fixed-point integral representations. This quantization technique can significantly reduce the storage requirements and computation of the model, thus improving inference speed [1]. This technique is particularly suitable for resource-constrained environments such as mobile devices, and in network architectures such as Google's Mobile Net, especially in resource-constrained environments such as mobile devices and embedded systems, models with fixed-point quantization have shown significant improvement in inference speed compared to floating-point models [2].

3. Pruning Techniques

To solve the sparsity problem of weighted pruning, structured pruning has emerged. Structured pruning compresses by removing the entire convolutional kernel, channel, or even layer, which allows the pruned model to still maintain structural integrity and facilitates efficient acceleration on hardware [3,4]. Various structured pruning algorithms have been integrated into Baidu's Paddle framework for optimizing the deployment of deep learning models in production environments [3]. The advantage of structured pruning is that it not only reduces the number of parameters in the model, but also effectively improves the inference speed, especially in embedded or mobile devices. However, the design and implementation of structured pruning is more complex. Pruning needs to ensure that the network structure is not damaged, to ensure that the model after pruning still has good training stability and inference accuracy.

4. Design and Optimization of Quantitative Methods

4.1. Selection of Quantization Bit Width

8-bit is the most common choice of quantization bit width because it has high computational efficiency in hardware gas pedals, while being able to guarantee the accuracy of most models [1]. However, in some applications that require less accuracy, lower bit widths can be chosen to further reduce the

computational effort [4]. Too low a bit width can lead to insufficient representation capability of the model, thus affecting the classification accuracy. Too high a bit width can reduce the accuracy loss, but it is difficult to fully utilize the advantages of quantization techniques.

4.2. QAT

Quantization-aware training (QAT) is a method that simulates quantization operations during training. Unlike direct quantization of a trained model, QAT introduces the simulation of quantization errors during the training phase, thus allowing the model to adapt to changes in accuracy after quantization during training [1,2]. We choose a layer-by-layer quantization strategy, where different network layers play different roles in the model, so that the quantization bit-width can be chosen to be higher at, for example, 8-bit for the convolutional layer, and lower for the fully connected layer. This can further reduce the computational effort of the model while minimizing the loss of accuracy.

5. Optimization Strategies for Quantitative Methods

We must design for inter-layer quantization differentiation. Different layers in a CNN have different sensitivities to quantization. In general, convolutional layers close to the input layer are more sensitive to quantization because these layers extract low-level image features, while fully connected layers close to the output layer are less sensitive to quantization. A balance between performance and accuracy can be achieved by using higher quantization accuracy in the sensitive layers and lower bit widths in the insensitive layers [1,2].

Zero offset in the quantization process is one of the key factors affecting the accuracy of the model. According to the distribution characteristics of the weights and activation values of each layer, the position of the zero point is dynamically adjusted so that the quantized value can better cover the actual data distribution and reduce the bias introduced by quantization [1].

To further improve the robustness of the quantization model, quantization techniques can be used in conjunction with regularization methods. During the training process, the regularization term is used to constrain the weight distribution after quantization [1,2], making the model more robust under fixed-point representation. This method can effectively reduce the overfitting phenomenon and improve the generalization ability of the model [4].

The ImageNet dataset is selected for experiments to evaluate the effect of quantization methods on model performance. The CNN architecture is chosen as the experimental object to analyze the effect of quantization methods by comparing the model performance before and after quantization. The metrics of model accuracy, inference speed, model size, and memory consumption after quantization are evaluated, focusing on the trade-off between accuracy loss and computational complexity.

Model accuracy, the ResNet-50 model has a Top-1 classification accuracy of 76.4% and a Top-5 accuracy of 93.2% on the ImageNet validation set. This is the best performance metric for the unquantized model. Inference speed, tested on a GPU (NVIDIA V100), the inference speed of the unquantized model is about 90 FPS (Frames Per Second), i.e., it can process 90 images per second. Model size, the storage requirement of the unquantized model is 98 MB, which is mainly determined by the number of parameters (about 25.5 million parameters). Memory consumption, during the test, the model occupied about 1.5 GB of video memory.

After 8-bit fixed-point quantization, the Top-1 accuracy of the model decreases to 74.8% and the Top-5 accuracy decreases to 92.0%. This indicates that quantization brings some accuracy loss, which is mainly due to the decrease in parameter accuracy during the quantization process, resulting in a decrease in the model's ability to capture subtle features. The inference speed of the quantized model on the same GPU increases to 150 FPS, an improvement of about 66%. This is due to the higher parallelism of 8-bit operations on the GPU, as well as the reduced computation of each weight and activation value, which significantly speeds up the inference of the model. The size of the 8-bit quantized model is about 25 MB, which is about a quarter of the original model. The significant reduction in the size of the quantized model means that it is more efficiently deployed in mobile devices and embedded systems, reducing storage requirements. The memory footprint of the quantized model drops to about 600MB, a reduction of about 60%. This is because the activation values and intermediate feature maps of each layer are quantized to 8-bits during the forward propagation process, significantly reducing memory requirements.

The 8-bit quantization brings about 1.6% Top 1 accuracy degradation, but this accuracy loss is acceptable in real application scenarios after model compression. For certain application scenarios with low accuracy requirements, the accuracy of the quantized model is sufficient to meet the needs.

The computational complexity of the quantized model is significantly reduced, and the inference speed is improved by 66%. It is useful for real-time applications in mobile devices, embedded systems, etc.

6. Optimization Strategies for Pruning Methods

The first step in choosing a pruning method is to cut out the convolutional kernels and selectively remove unimportant convolutional kernels, thus reducing the amount of computation and the number of parameters. Pruning channels removes the entire channel and can significantly reduce the amount of computation for the network, but make sure that the network is still able to effectively capture the features of the input image after pruning. Clipping layers can, in some cases, remove entire convolutional layers, but this usually significantly affects the feature extraction capability of the model, so special care is needed.

6.1. Iterative Pruning Process

The number of parameters and computation of the model will be significantly reduced after the initial pruning, but the accuracy may be significantly reduced. Retraining of the pruned model is performed to retune the model parameters using the training data from the unpruned time to restore the classification accuracy of the model [3]. A lower learning rate can be used for retraining to avoid drastic parameter changes in the model. The pruning and retraining process described above is repeated, with a portion of unimportant convolutional kernels or channels cut out each time and retrained. Through multiple iterations, the computational and parametric quantities of the model are gradually reduced

while keeping the accuracy loss within acceptable limits [5].

6.2. Hyperparameter Selection During Pruning

The proportion of convolutional kernels or channels removed at each pruning is an important hyperparameter in the pruning process. It can usually be started from 10% and gradually adjusted experimentally to find the optimal pruning ratio. The number of retraining sessions after pruning depends on the recovery of model accuracy. Usually, several complete rounds of retraining are required after pruning to recover the model accuracy [3,4].

6.3. Experimental Design and Evaluation of Effects

The ImageNet dataset is selected for the experiment and the benchmark model is ResNet-50. the original ResNet-50 model has a Top-1 accuracy of 76.4% and a Top-5 accuracy of 93.2% on the ImageNet validation set.

After several rounds of pruning and retraining, the model with a pruning ratio of 50% is able to control the Top-1 accuracy at about 74.0% and the Top-5 accuracy at about 91.5%. The size of the pruned model is reduced by about 50%, from 98MB to about 49MB, and the inference speed of the pruned model is improved by about 40% on GPU, indicating that structured pruning has obvious advantages in improving inference efficiency.

7. Combining Strategies

While quantization or pruning techniques alone can be effective in compressing models, a combination of the two can further optimize model performance. However, quantization faces challenges when combined with pruning.

Both quantization and pruning introduce a certain loss of accuracy, and the superposition of the two may lead to an excessive decrease in accuracy, making it difficult to meet the accuracy requirements of practical applications. Choosing the appropriate combination order is crucial to maximize the model performance [4]. The structure of the model changes after pruning, while quantization requires stable weight distribution in each layer, so the training stability of the model after pruning becomes a problem to be solved.

7.1. Design and Implementation of Combination Strategies

The first strategy is pruning followed by quantization, where unimportant convolutional kernels or channels are first removed from the model based on weight importance or redundancy analysis. This stage can be done by using structured pruning methods to remove redundant convolutional kernels, channels, or entire convolutional layers. The model accuracy usually decreases after pruning, so the pruned model needs to be retrained to restore accuracy. At this stage, the structure of the model is fixed, providing a basis for subsequent quantization. Quantization aware training of the pruned model. Since the pruned model is already more streamlined, QAT can further quantize the weights and activation values into low precision fixed-point representations, thus significantly reducing the amount of computation and storage requirements [1-4].

Then there is a post-quantization pruning strategy, which first performs preliminary quantization, quantization-aware training on the original model, and converts the weights and activation values of the model into fixed-point representations. This stage can reduce the computational complexity of the model and provide an optimization basis for subsequent pruning operations [1,2]. Based on the quantized model, unimportant convolutional kernels or channels are selectively removed based on the quantized weight distribution and activation value range [1]. Unlike traditional pruning, the weights of the model have been quantized at this point, and the accuracy of the quantized weights needs to be considered when pruning. The pruned model needs to be retrained to recover accuracy. At this point, the retrained model is already in its quantized form, and the quantization operation simulated during training makes the model more robust and better able to adapt to the loss of accuracy after quantization [1,3-4].

7.2. Experimentation and Effectiveness Evaluation of Combination Strategies

Experiments are conducted using the ImageNet dataset to evaluate the impact of the two combined strategies on the model performance. ResNet-50 is chosen as the benchmark model, and the original unpruned and unquantized model is first trained and its Top 1 and Top 5 accuracies are recorded. This includes accuracy loss (Top 1 and Top 5 accuracy), model size, inference speed, and memory usage.

7.2.1. Effects of Quantification after Pruning

Model size, the size of the model that is pruned and then quantized is reduced by about 75%, from 98MB in the original model to about 25MB.Inference speed, the inference speed of the model that is pruned and then quantized is increased by about 60%, which can significantly improve the inference efficiency. Accuracy, in terms of Top 1 accuracy, the model accuracy decreases from the original 76.4% to 74.5%, and Top-5 accuracy decreases from 93.2% to 91.0%. The accuracy loss is relatively controllable.

7.2.2. Quantifying the Effect of Post Pruning

Model size, quantization followed by pruning reduces the model size by about 70%, from 98MB in the original model to about 30MB. inference speed, quantization followed by pruning improves the inference speed of the model by about 50%, which is slightly less effective than the quantization strategy after pruning. Accuracy, Top 1 accuracy decreased from 76.4% to 73.0% and Top-5 accuracy decreased from 93.2% to 90.0%. The loss of accuracy is large, and the retraining process is difficult to fully recover.

7.2.3. Optimization Strategies and Future Research Directions

A joint optimization strategy is used to consider quantization and pruning simultaneously, design a joint loss function, and dynamically weigh the effects of both during the training process to maximize the model compression rate while minimizing the accuracy loss.

Explore the joint optimization strategy of quantization and pruning in multi-task learning for efficient compression and acceleration of multi-task models. Design adaptive quantization and pruning strategies based on the characteristics of different hardware platforms.

8. Conclusion

In this study, I explore the combined use of fixed-point quantisation and structured pruning techniques to optimize the performance and efficiency of convolutional neural networks (CNNs) in image classification tasks. By reducing model size and computational complexity, these approaches

make CNNs more suitable for deployment in resource-constrained environments such as mobile devices and embedded systems. Experimental results using ResNet-50 on the ImageNet dataset show that the combination of quantisation and pruning can reduce model size by up to 75% and increase inference speed by 50%, while maintaining an acceptable classification accuracy of 74.5%, compared to 76.4% for the baseline model. While the accuracy drops slightly, the increased efficiency makes this approach ideal for real-time applications. Future work will further focus on improving the combined quantisation and pruning strategies, as well as exploring their applicability to other CNN architectures and tasks.

References

[1] Azarpeyvand, A., Rokh, B., & Naderi, S. (2023). A comprehensive survey on model quantization for deep neural networks in image classification. ACM Transactions on Intelligent Systems and Technology, 14(6), 1-50.

[2] Cheng, Y., Wang, X., Xie, X., Li, W., & Peng, S. (2022). Channel pruning guided by global channel relation. Applied Intelligence, 42(5), 1234-1250.

[3] Dai, B., Zhu, C., Guo, B., & Wipf, D.P. (2018) An adaptive joint optimization framework for pruning and quantization. In: 35th International Conference on Machine Learning. Stockholm. pp. 1143-1152.

[4] Li, Z., Li, H., & Meng, L. (2023). A survey on deep neural network pruning. Computers, 12(3), 60.

[5] Zacchigna, G. F., Lew, S., & Lutenberg, A. (2024). Flexible quantization for efficient convolutional neural networks. Electronics, 13(10), 1923.