BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2023
http://excel.fit.vutbr.cz

Automated Quantization of Neural Networks

Miroslav Šafář*

Abstract

Quantization of deep neural networks is a common way to optimize memory and energy requirements for deployment on energy and memory-constrained devices while maintaining an acceptable accuracy loss. Mixed-precision quantization allows even better memory and energy savings. But selecting the precision for each layer needs expert knowledge and a deep analysis of the network. In this paper, we address this problem and we present a system for automated mixed-precision quantization of neural networks. We utilize the multi-objective evolutionary algorithm NSGA-II and quantization-aware training for fine-tuning the quantization configurations. We conducted experiments with a tiny-imagenet dataset and MobileNetv1 0.25. We achieved accuracies comparable to those of floating-point models while making the model about 40 % smaller in comparison to 8-bit models. This beats uniform quantization by 10 %. These size savings can be used to save the costs of memories in the hardware accelerators during manufacturing or allow deployment of deep neural networks to even smaller devices.

*xsafar23@fit.vutbr.cz, *Faculty of Information Technology, Brno University of Technology*

1. Introduction

Deep neural networks (DNNs) have played an important role in artificial intelligence in the past decade. They have allowed the huge advance in computer vision and speech recognition and now they are moving to the edge. Mobile devices powered by AI have become the standard and AI-powered IoT nodes and embedded devices are going to be next.

But it is not an easy task to deploy big DNNs with tens of millions of parameters to small energy and memory-constrained devices like embedded ones. Smaller optimized models started emerging and research started focusing on DNN optimization. One of these techniques is a quantization of network parameters, which can lead to just a small accuracy drop for moving from floating point to 8-bit integers and a huge benefit in memory size and energy consumption.

This compression can be taken further and quantization can go as low as 1-bit. But compression to this level leads to a big accuracy loss. Mixed-precision quantization addresses this problem, where important layers are quantized with higher precision while the insignificant ones are quantized to a much lower level, but deciding the bit precision of each layer needs expert-level skills. My work addresses that and proposes a system to automatically search for the best mixed-precision quantization configurations.

There is similar work [1] proposing its framework for searching mixed-precision quantization configurations based on reinforcement learning. But it relies on framework hardware support and gives the user the best-found solution. In contrast, we propose using the multi-objective evolutional algorithm NSGA-II to search the configuration space and provide the user with Pareto-optimal solutions to choose the best trade-off for them. Also, we focus on the usage of quantization-aware training [2] to achieve better quantization accuracy of smaller models. To analyze the effect of different quantization techniques, we have implemented support for per-layer quantization into the TensorFlow library.

We have evaluated the proposed solution on Mobilenet 0.25 with a tiny-imagenet dataset, which consisted of 100 classes with 1000 samples for training and all original validations samples for these classes for validation. We achieved accuracies comparable to those of floating-point models while making the model about 30 % smaller for per-tensor asymmetric quantization and 40 % smaller for per-channel symmetric quantization in comparison to 8-bit models. This beats uniform quantization by 10 %, respectively by 6 % for per-channel weights quantization.

2. Architecture of proposed system

The overall architecture of the proposed system is presented in Figure 1. The input of the system is a pre-trained TensorFlow model. Then the number of quantizable layers is calculated with batch normalization layer folding in mind. Then the NSGA-II evolutionary algorithm is utilized and after a given number of generations, the best-found solutions are fine-tuned and are the output of the proposed system.

2.1 Mixed-precision quantization in TensorFlow

To allow basic chromosome representation of layer quantization configuration we have designed a function that quantizes the model with layer quantize configuration without any other requirements. To do so, we have utilized ModelTransformer and DefaultNBitQuantizeRegistry provided by the TensorFlow library. We developed PerLayerQuantizeModelTransormer that splits model layers into independent groups using BFS graph search and allows separate quantization configurations for each group. We also developed PerLayerQuantizeRegistry which accepts layer configuration and then provides configuration classes to the TensorFlow library's `quantize_apply` function.

2.2 Evolutionary algorithm NSGA-II

Quantization configurations are represented by chromosomes where each gene is from 2 to 8, representing the bit precision of one layer of the original model. The initial population consists of all uniform solutions and their offspring. Individuals in the population are evaluated by applying per-layer quantization configuration on the input model. A fake quantized model is then partially fine-tuned using quantization-aware training for a few epochs (12 epochs were used for experiments) while validation accuracy is evaluated after each and the best one is taken. We also calculate weights memory size in the quantized model.

Parents for the new population are selected using nondominated fronts as described in [3]. We have a 10 % chance for mutation where one gene of the chromosome is changed to a random N-bit precision.

After a given number of generations, the best nondominated front is chosen and configurations are fully fine-tuned using 50 epochs of quantization-aware training.

3. Experiments and Results

We have experimentally validated the proposed system. We have used a subset of Imagenet with 100 classes and with 1000 samples per class for training and all validation samples from these classes for validation. We have used Mobilenet 0.25 we trained in floating-point for 250 epochs and achieved 51.6 % Top-1 classification accuracy on our Imagenet subset. Then we run experiments with the proposed solution with two types of quantization – per-tensor asymmetric quantization and per-channel symmetric quantization. The time budget for NSGA runs was 24 hours on 8 NVIDIA A100 GPUs. Then the fine-tuning of found configurations was done. There are results in Figure 3 and Figure 4 for per-tensor asymmetric quantization and in Figure 5 and Figure 6 for per-channel symmetric quantization. The proposed system achieved a 10 % smaller model than a uniform solution with per-tensor asymmetric quantization while maintaining the same accuracy as the original floating-point model, as shown in Figure 4. Results are not so good for per-channel symmetric quantization, which is probably caused by a small number of epochs that could be computed in 24 hours. Nevertheless, in both experiments, the proposed system found better Pareto optimal solutions than uniform ones.

4. Conclusions

I am proud to say I have achieved the set goal with the results of this work. If I had more time, I would focus more on post-training quantization techniques to optimize the network before quantization-aware training to achieve better results in fewer epochs, so more generations could be done. Also, this work was part of university research and it is planned to add more evaluation parameters than just the memory size of weights.

Acknowledgements

I would like to thank my supervisor Ing. Vojtěch Mrázek Ph.D. for his help. This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic through the e-INFRA CZ (ID:90140). This work was supported by Czech science foundation project GA22-02067S.

References

[1] Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han. Haq: Hardware-aware automated quantization with mixed precision. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8604–8612, 2019.

[2] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference, 2017.

[3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.