MLE Quantization of Convolutional Neural Networks

Authors:

Natan Kaminsky, Almog Tzabari

Abstract

In recent years deep convolutional neural networks have become a key component in SOTA image processing techniques. But these networks require extensive computing resources due to the large number of parameters they have. An active research area has emerged to try solving this problem in various ways such as pruning the number of parameters in the neural networks or quantizing the parameters into fewer bits. Often after quantization or pruning, recalibration of the networks is required for recovering their degradation in performance. We propose a new post training quantization method for neural networks that can be applied without the training dataset or fine-tuning the network. We perform non-uniform quantization according to analytical analysis assuming a Laplace distribution. Neural networks quantized by our method show better or comparable performance to other SOTA 4-bit post training quantization methods.

Introduction

The need for more efficient neural networks has grown as deep learning models started to dominate a wide range of technologies. Mobile or embedded devices that have limited hardware struggle with running deep learning models due to the extensive computations required. This limits the practical usage of deep learning models in various applications. It has been shown that at 8-bit precision the networks retain accuracy close to their floating-point precision counterpart.[1] But a more aggressive quantization causes a significant drop in accuracy.[2] As a result, several papers were published, proposing different methods to improve efficiency.

One popular approach is to consider quantization while training the neural network. This method was suggested by Zhou et al.[3] and Choi et al.[4] With quantization-aware training the model can perform well up to 2-bit precision. However, this method requires training the deep learning model from scratch which is time consuming. Furthermore, the training dataset is not always accessible, which makes retraining not possible.

Another approach is to quantize a pretrained neural network without any retraining or fine-tuning. Zhao et al.[5] suggest splitting the network channels and halving their value to lower the values of outliers that increase the quantization range. This method incurs a significant overhead for calculations and does not work well for activations.
Nagel et al.[6] deal with outliers by equalizing the weight ranges in the network by making use of a scale-equivariance property of activation functions. In contrast to our method, Nagel et al. deal with quantization of bit widths higher than 4-bit precision.

Migacz[7] proposes clipping the activations in a manner that minimizes the KLD (KL divergence) between the floating-point distribution and quantized distribution. Finding the clipping values is an iterative process that requires a small calibration dataset.
Banner et al.[8] suggest clipping the activations as well by approximating the optimal clipping value analytically - ACIQ. This method finds the value that minimizes the MSE assuming in advance a Laplace or Gaussian distribution of the activations. Our non-uniform quantization scheme displays better or comparable results to the KLD method and ACIQ on various deep learning models.

Methods

It has been observed that activations are more sensitive to quantization than weights.[3] To minimize the loss of information we propose to non-uniformly quantize the activations in an efficient manner. We split the quantization bins into two groups: Densely spaced bins and sparsely spaced bins. The ratio between the bin width of the first group and second group will be denoted as β.

We define the relative error between the original tensor and its quantized version as $RelativeError(x, x_q) \triangleq \frac{|x - x_q|}{|x|},\ x > 0$

And therefore:

$RelativeError(x, x_q) \leq \alpha$ iff $x \in \left[\frac{x_q}{1+\alpha}, \frac{x_q}{1-\alpha}\right]$ for $0 < \alpha < 1$. The probability of an error smaller than α is:

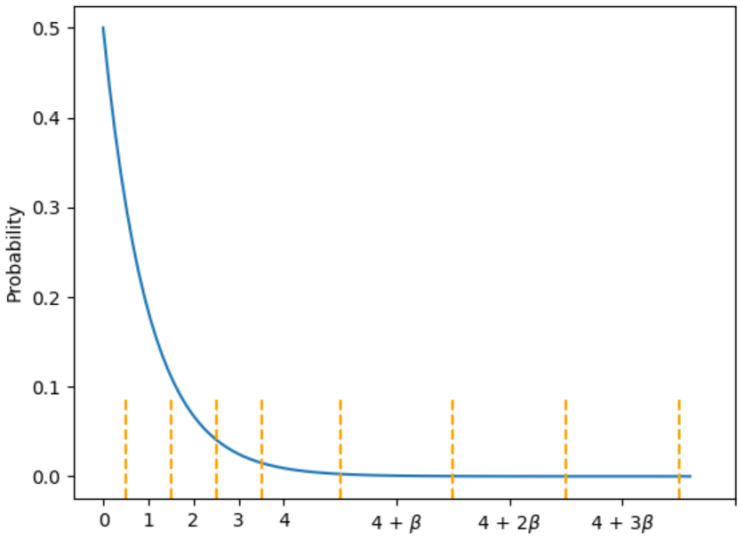

Figure 1: An illustration of non-uniform quantization with 3-bit precision of a rectified $Laplace(0,1)$ distribution. The densely spaced bins are placed where most of the tensor elements reside. On the other hand, the sparsely spaced bins are placed near the outliers. In practice, the bin locations are normalized by $\frac{S}{4+3\beta}$. β is the ratio between the widths of the dense bins and sparse ones, and S is the maximum value of the tensor.

$P\left(RelativeError(x, x_q) \leq \alpha \mid \max(x) = S\right) = P\left(x \in \cup_{r \in X_q}\left[\frac{r}{1+\alpha}, \frac{r}{1-\alpha}\right]\right)$ where X_q is the set of the quantization bin locations. For N-bit precision we can define the set X_q as:

$$X_q \triangleq \frac{S}{2^{N-1} + (2^{N-1} - 1)\beta}\left(\{0,1,\dots 2^{N-1} - 1\} \cup \{2^{N-1}, 2^{N-1} + \beta, 2^{N-1} + 2\beta, \dots 2^{N-1} + (2^{N-1} - 1)\beta\}\right)$$

To maximize the probability of an error smaller than α we solve for:

$$\underset{\beta}{\operatorname{argmax}}\, P\left(RelativeError(x, x_q) \leq \alpha\right)$$

In our calculations we set $\alpha = 0.01, N = 4$ and assume that each tensor has 25,000 elements independently sampled from a Laplace distribution. We can derive an analytical expression for the average S:

S is defined as $max\{X_1, X_2, \dots X_n\}$ for a tensor with n elements. Assuming all samples are independently sampled from the same distribution, its CDF and PDF are $F_X(x)^n$ and $nF_X(x)^{n-1}f_X(x)$ respectively. Therefore:

$$E(S) = \int_0^\infty nF_X(x)^{n-1}f_X(x)x dx$$

We solve this integral numerically for Laplace distribution.

In the preceding formulations the zero bin is not considered even though there might be a small number of elements correctly positioned there. In practice, the use of relative error is probably too strict for the zero bin and the quantization error of many small values there will not degrade the classification results. Therefore, we consider the zero bin in our calculations and give it a smaller weight (it gets multiplied by a small constant 0.1).

We approximate β by finding the optimal β which is most frequent over many tensors with elements independently sampled from a Laplace distribution.

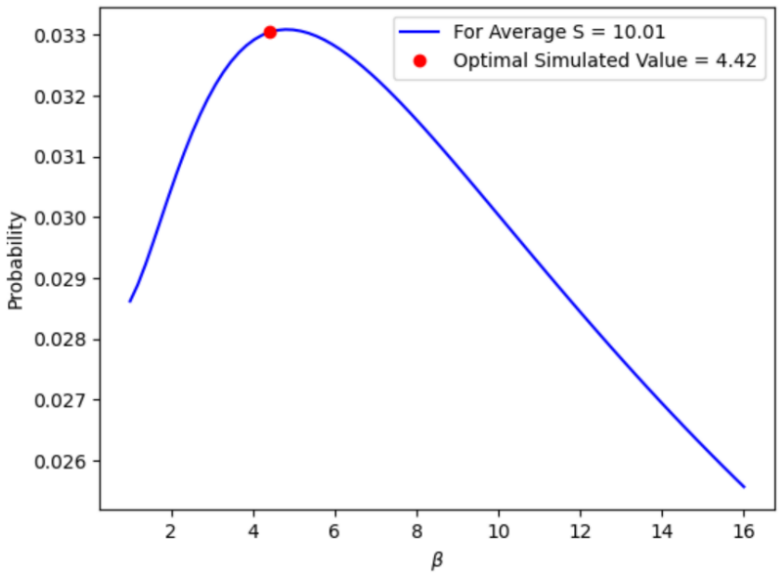

Figure 2: A graph of the probability of getting a smaller relative error than 1% assuming a $Laplace(0,1)$ distribution and a fixed S equal to the average maximum of a tensor with 25,000 elements. The optimal simulated value is not exactly at the peak of the graph because fixing S to the average maximum is an approximation for solving for β.

As we mentioned earlier, we want the non-uniform quantization process to be efficient from a hardware perspective. Therefore, we decided to restrict β to be a power of 2 (this will require only simple extra shift operations for multiplication). According to our simulations the solution for β is approximately 4.5, thus we set β to be 4 when carrying out non-uniform quantization.

Implementation and experiments

We quantized 6 different deep learning models (VGG16, VGG16-BN, ResNet-18, ResNet-50, ResNet-101, Inception v3) pretrained on the Imagenet dataset (ILSVRC 2012) with our method. Afterwards, we compared their accuracy to the KLD method and ACIQ accuracies reported in Banner et al.[8]. The pretrained models were obtained from the Torchvision library. These pretrained models were used for the KLD method and ACIQ previously as well.

The pretrained models expect the input to be mini-batches of 3-channel RGB images of shape (3 x H x W), where H and W are expected to be at least 224 (Inception v3 expects size 299). The images must be in a range of [0, 1] and then normalized to mean = [0.485, 0.456, 0.406] and std = [0.229, 0.224, 0.225]. We first resized each image to 256 (or 299), cropped its middle to size 224 and then normalized each channel mean and std to the values mentioned beforehand.

In this experiment the quantization of all methods was for the whole tensor and not per-channel (only one zero point and scale per tensor). We measured the quantized model's accuracies over the whole validation set (50,000 images) with a batch size of 1. 8 bits were allocated for the model's weights, and 4 bits for the activations. We applied

our non-uniform quantization with $\beta = 4$ only to the output of ReLUs. Otherwise, we performed regular asymmetrical quantization (with the preservation of zero) to the tensor.

The code for the experiments and its documentation can be found on:

https://github.com/nkami/mle_quantization

Results and discussion

The following table shows the accuracies of all the different deep learning models we tested.

Model	Naïve (8W4A)	KLD (8W4A)	ACIQ (8W4A)	Non-uniform (8W4A)	Reference (float32)
VGG16	53.90%	67.04%	**67.40%**	67.13%	71.59%
VGG16-BN	29.50%	65.85%	67.60%	**67.84%**	73.36%
ResNet-18	53.20%	65.06%	**65.80%**	**65.80%**	69.75%
ResNet-50	52.70%	70.80%	**71.45%**	71.05%	76.10%
ResNet-101	50.80%	71.70%	69.53%	**72.20%**	77.30%
Inception v3	41.40%	59.25%	60.80%	**65.44%**	77.20%

Figure 3: As mentioned in Banner et al. work, the naïve column refers to regular asymmetrical quantization (between minimum and maximum values). The KLD and ACIQ columns refer to the KLD method and ACIQ accuracies, respectively. The non-uniform refers to our quantization method.

Conforming to previous work, the Naïve quantization causes a significant drop in accuracy and gets outclassed by all other methods. Non-uniform quantization produces better results than the KLD method in all the deep learning models. Furthermore, the KLD method is an iterative process which is much slower than any of the other methods. ACIQ performed the best for VGG16 and ResNet-50 and tied for ResNet-18 with non-uniform quantization. However, our method is a close second in all these models. The non-uniform quantization method gets the best results for VGG16-BN, ResNet-101 and Inception v3. More specifically, we see significant improvements of 2.67% and 4.64% in the accuracies of ResNet-101 and Inception v3 compared to ACIQ.

In future work it may be possible to include non-uniform quantization to the weights (assuming a symmetrical distribution) for lower bit widths as well. Another interesting avenue of research could be to combine more advanced clipping methods with the non-uniform quantization scheme. Currently, quantized deep learning models are not common in practical applications because of the downsides of training-aware quantization such as dataset unavailability or the time-consuming training process. In this work we present a new way to perform post training quantization that shows better or comparable results to other SOTA post training quantization techniques. We hope this new quantization scheme can pave the way for new research and applications in the industry.

References

1. Raghuraman Krishnamoorthi, Quantizing deep convolutional networks for efficient inference: A whitepaper, 2018. https://arxiv.org/pdf/1806.08342.pdf

2. Jacob, Kligys, Chen, Zhu, Tang, Howard, Adam and Kalenichenko, Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference, 2017. https://arxiv.org/pdf/1712.05877.pdf

3. Zhou, Wu, Ni, Zhou, Wen, Zou, DOREFA-NET: Training Low Bit Width Convolutional Neural Networks with Low Bit Width Gradients, 2018. https://arxiv.org/pdf/1606.06160.pdf

4. Choi, Wang, Venkataramani, I-Jen Chuang, Srinivasan, Gopalakrishnan, PACT: Parametrized Clipping Activation for Quantized Neural Networks, 2018. https://arxiv.org/pdf/1805.06085.pdf

5. Zhao, Hu, Dotzel, De Sa, Zhang, Improving Neural Network Quantization without Retraining using Outlier Channel Splitting, 2019. https://arxiv.org/pdf/1901.09504.pdf

6. Nagel, Baalen, Blankevoort, Welling, Data-Free Quantization Through Weight Equalization and Bias Correction, 2019. https://arxiv.org/pdf/1906.04721.pdf

7. Migacz, 8-bit Inference with TensorRT, 2017. https://on-demand.gputechconf.com/gtc/2017/presentation/s7310-8-bit-inference-with-tensorrt.pdf

8. Banner, Nahshan, Soudry, Post training 4-bit quantization of convolutional networks for rapid-deployment, 2019. https://arxiv.org/pdf/1810.05723.pdf