Channel Balancing for Accurate Quantization of Winograd Convolutions Supplementary Material

Vladimir Chikin
Huawei Noah's Ark Lab
vladimir.chikin@huawei.com

Vladimir Kryzhanovskiy
Huawei Noah's Ark Lab
kryzhanovskiy.vladimir@huawei.com

A. Winograd transformation matrices

In the paper we presented the Balanced Quantized Winograd formula by parts to simplify the discussion. The formula for the most complex case, i.e., the tile-wise quantization scales $\mathbf{s}_v \in \mathbb{R}^{a\times a}$ and $\mathbf{s}_u \in \mathbb{R}^{a\times a}$, has the following form:

$$\mathbf{Y}_f = \mathbf{A}^T \frac{\sum_c^C \tilde{\mathbf{V}}_c \odot \tilde{\mathbf{U}}_{fc}}{\mathbf{s}_u \odot \mathbf{s}_v}\mathbf{A}, \tag{A.1}$$

where

$$\tilde{\mathbf{V}}_c = \lfloor \mathbf{B}^T\mathbf{X}_c\mathbf{B} \odot (\mathbf{s}_v \odot \tfrac{1}{\mathbf{\Omega}_c}) \rceil \tag{A.2}$$

and

$$\tilde{\mathbf{U}}_{fc} = \lfloor \mathbf{G}\mathbf{W}_{fc}\mathbf{G}^T \odot (\mathbf{s}_u \odot \mathbf{\Omega}_c) \rceil. \tag{A.3}$$

Here $\odot$ is the element-wise Hadamard product. $\mathbf{X}_c \in \mathbb{R}^{a\times a}$ is the c-th channel of a sub-tensor, obtained by sliding a window of the size $a \times a$ with stride m over the feature map. Hereinafter $a = m+k-1$ ($k \equiv 3$) and $c = 1, 2, ..., C$ (C (capital letter) is the number of channels of the feature map); $\mathbf{B} \in \mathbb{R}^{a\times a}$, $\mathbf{G} \in \mathbb{R}^{a\times k}$, and $\mathbf{A} \in \mathbb{R}^{a\times m}$ are data, weights, and inverse transformation matrices respectively (see their definitions below); $\mathbf{\Omega}_c \in \mathbb{R}^{a\times a}$ is the balancing matrix ($\mathbf{s}_v \odot \frac{1}{\mathbf{\Omega}_c}$ means that the matrix $\mathbf{s}_v$ is divided element-wise by the matrix $\mathbf{\Omega}_c$); $\tilde{\mathbf{V}}_c, \tilde{\mathbf{U}}_{fc} \in \mathbb{Z}^{a\times a}$ are an integer matrices, obtained by applying quantization operator $\lfloor\cdot\rceil$:

$$\lfloor \mathbf{Z} \cdot \mathbf{s} \rceil = \mathbf{Clip}\Big(\mathbf{Round}\big(\mathbf{Z} \cdot \mathbf{s}\big), -n, +n\Big), \tag{A.4}$$

using quantization scales $\mathbf{s}_v$ and $\mathbf{s}_u$, $n = 2^{b-1} - 1$, b is the quantization bitwidth. Finally, $f = 1, 2, ..., F$ denotes the filter number.

Winograd convolution algorithms can be constructed using Lagrange interpolation, and therefore their specific implementations depend on the choice of interpolation parameters, see [2] or [3]. In our paper, we use the best-known versions of the Winograd algorithm introduced in [2]. For F(4, 3) ($m = 4, a = 6$) Winograd algorithm, the Winograd transformation matrices are:

$$\mathbf{B}^T = \begin{pmatrix} 4 & 0 & -5 & 0 & 1 & 0 \\ 0 & -4 & -4 & 1 & 1 & 0 \\ 0 & 4 & -4 & -1 & 1 & 0 \\ 0 & -2 & -1 & 2 & 1 & 0 \\ 0 & 2 & -1 & -2 & 1 & 0 \\ 0 & 4 & 0 & -5 & 0 & 1 \end{pmatrix} \tag{A.5}$$

$$\mathbf{G} = \begin{pmatrix} \frac{1}{4} & 0 & 0 \\ -\frac{1}{6} & -\frac{1}{6} & -\frac{1}{6} \\ -\frac{1}{6} & \frac{1}{6} & -\frac{1}{6} \\ \frac{1}{24} & \frac{1}{12} & \frac{1}{6} \\ \frac{1}{24} & -\frac{1}{12} & \frac{1}{6} \\ 0 & 0 & 1 \end{pmatrix} \tag{A.6}$$

$$\mathbf{A}^T = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & -1 & 2 & -2 & 0 \\ 0 & 1 & 1 & 4 & 4 & 0 \\ 0 & 1 & -1 & 8 & -8 & 1 \end{pmatrix} \tag{A.7}$$

For F(6, 3) ($m = 6, a = 8$) Winograd algorithm, the Winograd transformation matrices have the following form, see [1]:

$$\mathbf{B}^T = \begin{pmatrix} 1 & 0 & -\frac{21}{4} & 0 & \frac{21}{4} & 0 & -1 & 0 \\ 0 & 1 & 1 & -\frac{17}{4} & -\frac{17}{4} & 1 & 1 & 0 \\ 0 & -1 & 1 & \frac{17}{4} & -\frac{17}{4} & -1 & 1 & 0 \\ 0 & \frac{1}{2} & \frac{1}{4} & -\frac{5}{2} & -\frac{5}{4} & 2 & 1 & 0 \\ 0 & -\frac{1}{2} & \frac{1}{4} & \frac{5}{2} & -\frac{5}{4} & -2 & 1 & 0 \\ 0 & 2 & 4 & -\frac{5}{2} & -5 & \frac{1}{2} & 1 & 0 \\ 0 & -2 & 4 & \frac{5}{2} & -5 & -\frac{1}{2} & 1 & 0 \\ 0 & -1 & 0 & \frac{21}{4} & 0 & -\frac{21}{4} & 0 & 1 \end{pmatrix} \tag{A.8}$$

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{2}{9} & -\frac{2}{9} & -\frac{2}{9} \\ -\frac{2}{9} & \frac{2}{9} & -\frac{2}{9} \\ \frac{1}{90} & \frac{1}{45} & \frac{2}{45} \\ \frac{1}{90} & -\frac{1}{45} & \frac{2}{45} \\ \frac{32}{45} & \frac{16}{45} & \frac{8}{45} \\ \frac{32}{45} & -\frac{16}{45} & \frac{8}{45} \\ 0 & 0 & 1 \end{pmatrix} \tag{A.9}$$

Table A.1. QAT of ResNet-20 with Winograd convolutions on the CIFAR-10 dataset: the employed set of learning rate schedules.

Number of LR schedule	Learning Rate Schedule
1	From beginning: LR $= 10^{-5}$.
2	From beginning: LR $= 10^{-4}$, from 50 epoch: LR $= 10^{-5}$.
3	From beginning: LR $= 10^{-3}$, from 50 epoch: LR $= 10^{-4}$.
4	From beginning: LR $= 10^{-2}$, from 100 epoch: LR $= 10^{-3}$.
5	From beginning: LR $= 10^{-2}$, from 100 epoch: LR $= 10^{-3}$, from 200 epoch: LR $= 10^{-4}$.

Table A.2. Inference time for $F(4,3)$ Winograd algorithm, $H = W = 128$, 8-bit quantization. Measured on CPU: Intel(R) Core(TM) i7-7700 CPU @ 4.10GHz, one thread.

F=C	Static, ms				Dynamic, ms			
	QW	BQW	Overhead	BQW fused	QW	BQW	Overhead	Overhead (theory)
8	34	35	4.5%	34	41	43	4.7%	2.4%
16	71	75	5.0%	71	83	87	4.7%	2.3%
32	163	169	3.7%	163	182	190	4.0%	2.0%
64	393	408	3.8%	393	441	456	3.4%	1.6%
128	1066	1093	2.5%	1070	1187	1209	1.8%	1.0%
256	3199	3256	1.8%	3209	3475	3516	1.2%	0.5%
512	10733	10803	0.6%	10734	11216	11328	1.0%	0.2%

$$\mathbf{A}^T = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & -1 & 2 & -2 & \frac{1}{2} & -\frac{1}{2} & 0 \\ 0 & 1 & 1 & 4 & 4 & \frac{1}{4} & \frac{1}{4} & 0 \\ 0 & 1 & -1 & 8 & -8 & \frac{1}{8} & -\frac{1}{8} & 0 \\ 0 & 1 & 1 & 16 & 16 & \frac{1}{16} & \frac{1}{16} & 0 \\ 0 & 1 & -1 & 32 & -32 & \frac{1}{32} & -\frac{1}{32} & 1 \end{pmatrix} \quad \text{(A.10)}$$

B. Training configurations for Winograd QAT

We train ResNet-20 on the CIFAR-10 dataset with quantized Winograd convolutions using the SGD optimizer with a momentum of 0.9 and weight decay of 10^{-6}. In all our experiments, the batch size is 100. We train the models using several different learning rate schedules for 300 epochs for each considered Winograd algorithm and bitwidth, for both balanced quantized Winograd (BQW) convolutions and traditional quantized Winograd (QW) convolutions. These learning rate schedules are listed in Tab. A.1. We consider schedules, in which learning rates of various orders gradually decrease during training. For both BQW and traditional QW, we provide the best result obtained in several training runs using the learning rate schedules from Tab. A.1.

In all experiments, we use 100 batches from the training dataset to calibrate the balancing coefficients, as well as to estimate the quantization parameters of the layer inputs of the Winograd convolutions. We use such a sufficiently large amount of data in order to make the experimental results be less dependent on the data used for initialization of balancing coefficients and input quantization parameters. However, in practice a much smaller amount of data could be utilised to initialize high-quality balanced quantized Winograd convolutions.

C. Inference time measurements

We implemented BQW and QW convolutions in C language and measured their inference time for the case of 8-bit quantization (see Tab. A.2). Unfortunately, we did not have opportunity to make low-level optimizations (such as vectorization by intrinsic function, using Assembler, etc.). The real overhead strongly depends on efficiency of the implementation. Nevertheless, the inference time ratio between BQW and QW demonstrates the overhead nature of the channel balancing operation. Fig. C.1 shows the experimental and theory overheads for the case of dynamic quantization. We observe that the curves related to theory and experiments have a similar behavior. The findings of the paper are confirmed by the real inference time measurements:

1. For both static and dynamic quantization the overhead of the channel balancing is small and decreases with the increasing number of filters and channels. Please see the "Overhead" columns, which demonstrate the relative increase of the BQW inference time relative to the QW inference time: $overhead = BQW/QW - 1$.

2. In the case of static quantization, the balancing coeffi-

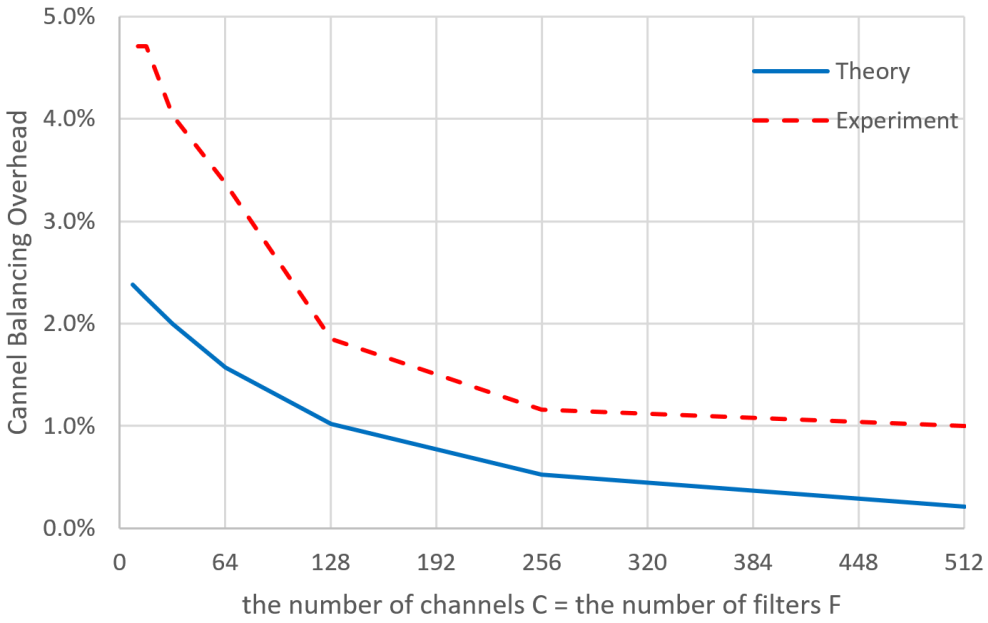

Figure C.1. The overhead of the channel balancing operation for the case of dynamic quantization. Data were taken from Tab. A.2.

cients can be fused into the quantization scale. Hence, the overhead becomes close to zero (compare "BQW" and "BQW fused" columns).

References

[1] Andrew Lavin. Wincnn. https://github.com/andravin/wincnn, 2020. 1

[2] Andrew Lavin and Scott Gray. Fast algorithms for convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4013–4021, 2016. 1

[3] Lingchuan Meng and John Brothers. Efficient winograd convolution via integer arithmetic. *arXiv preprint arXiv:1901.01965*, 2019. 1