ShiftCNN: Generalized Low-Precision Architecture for Inference of Convolutional Neural Networks

2018-10-19

Overview

- 2017
- 方法：ShiftCNN
- 作者： Denis A. Gudovskiy, Luca Rigazio
- 优点：
  - 1 降低了卷积的计算成本，使用编码表将原本的weight编码成2^n，将原本乘法运算x*w变为x>>1+x>>2+x>>3+...这样的形式
  - 2 给出了完整的ShiftCNN代码和模块设计
  - 3 指出这一设计可以在FPGA上加速运算

ShiftCNN Quantization

$$\hat{w}_i = \sum_{n=1}^{N} \mathcal{C}_n[\mathrm{idx}_i(n)],$$ 目的：把w转化为如左形式表示

where for ShiftCNN codebook set $\mathcal{C}_n = \{0, \pm 2^{-n+1}, \pm 2^{-n}, \pm 2^{-n-1}, \ldots, \pm 2^{-n-\lfloor M/2 \rfloor+2}\}$ and a codebook entry can be indexed by B-bit value with total $M = 2^B - 1$ combinations. Then, each weight entry can be indexed by (NB)-bit value. For example, consider a case with $N = 2$ and $B = 4$. Then, the weight tensor is indexed by 8-bit value with codebooks $\mathcal{C}_1 = \{0, \pm 2^0, \pm 2^{-1}, \ldots, \pm 2^{-6}\}$ and $\mathcal{C}_2 = \{0, \pm 2^{-1}, \pm 2^{-2}, \ldots, \pm 2^{-7}\}$. Note that for $N = 1$, (5) can be simplified to binary-quantized ($B = 1, 0 \notin \mathcal{C}_1$) model or ternary-quantized ($B = 2$) model. Moreover, fixed-point integers can be represented by (5) as well when $N \gg 1$. Similar to (Zhou et al., 2017), we normalize **W** by

Cn：编码表，N个编码表，每个里面存2^n-1个数
每个weight可以使用NB-bit索引查找对应值

$$\mathcal{C}_n = \{0, \pm 2^{-n+1}, \pm 2^{-n}, \pm 2^{-n-1}, \ldots, \pm 2^{-n-\lfloor M/2 \rfloor+2}\}$$

$N = 2$ and $B = 4$.

$$\mathcal{C}_1 = \{0, \pm 2^0, \pm 2^{-1}, \ldots, \pm 2^{-6}\}$$

$$\mathcal{C}_2 = \{0, \pm 2^{-1}, \pm 2^{-2}, \ldots, \pm 2^{-7}\}.$$

1. Bit数低
2. 加速运算
3. 空间换时间（bit数低的情况下存储尽可能增加精度，O(N)，实际只有P=M+2(N−1)个不同编码）

ShiftCNN Quantization

用编码表和index对weight做近似，目的是加速卷积运算

Algorithm 1 Quantization to ShiftCNN weight representation

Initialize: $\tilde{w}_i = 0$, $r = w_i / \max(\text{abs}(\mathbf{W}))$, where $i \in \{\tilde{c}, c, h_f, w_f\}$
for $n = 1 \ldots N$ **do**
    $q_{\text{sgn}} = \text{sgn}(r)$
    $q_{\log} = \log_2 |r|$
    $q_{\text{idx}} = \lfloor q_{\log} \rfloor$
    $b_{\log} = q_{\text{idx}} + \log_2 1.5$
    if $q_{\log} > b_{\log}$ **then**
        q_{idx}++
    $q = q_{\text{sgn}} 2^{q_{\text{idx}}}$
    $\text{idx}_i(n) = q_{\text{sgn}}(-n - q_{\text{idx}} + 2)$
    if $|\text{idx}_i(n)| > \lfloor M/2 \rfloor$ **then**
        $q = 0$
        $\text{idx}_i(n) = 0$
    r -= q
    $\tilde{w}_i$ += q

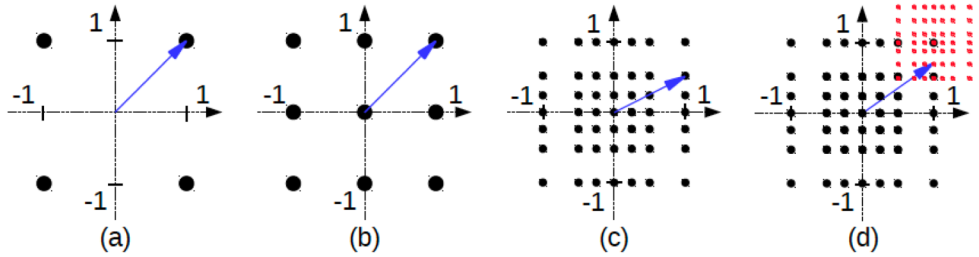

Figure 1: Weight vector in 2D space for models: (a) binary-quantized ($N = 1, B = 1, 0 \notin \mathcal{C}_1$), (b) ternary-quantized ($N = 1, B = 2$), (c) ($N = 1, B = 3$) and (d) ($N = 2, B = 3$).

ShiftCNN Architecture

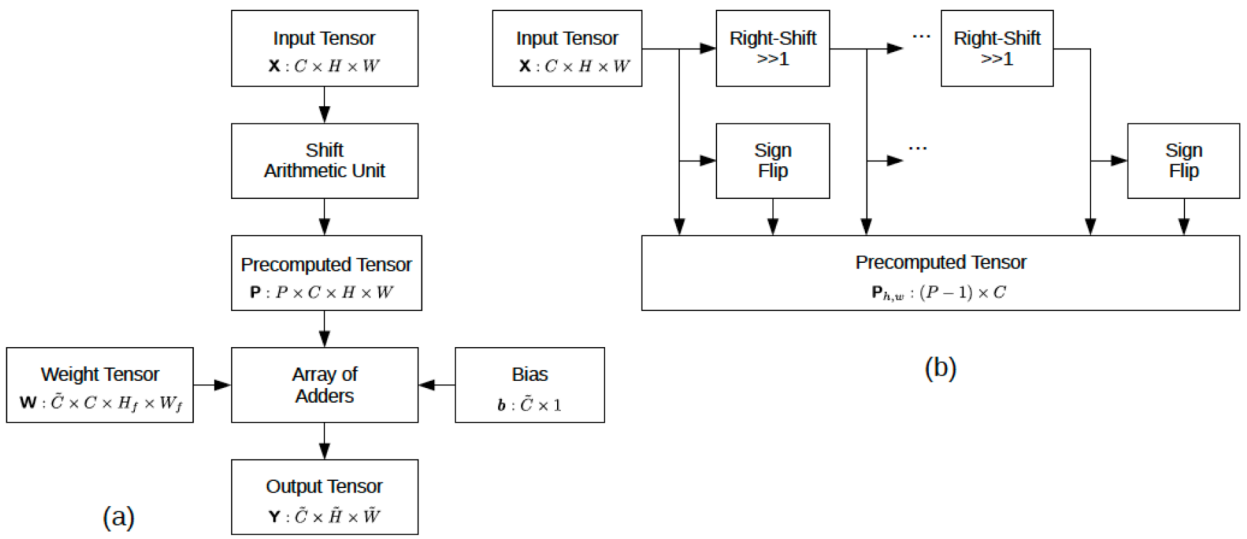

Figure 2: (a) High-level structure. (b) Shift Arithmetic Unit (ShiftALU).

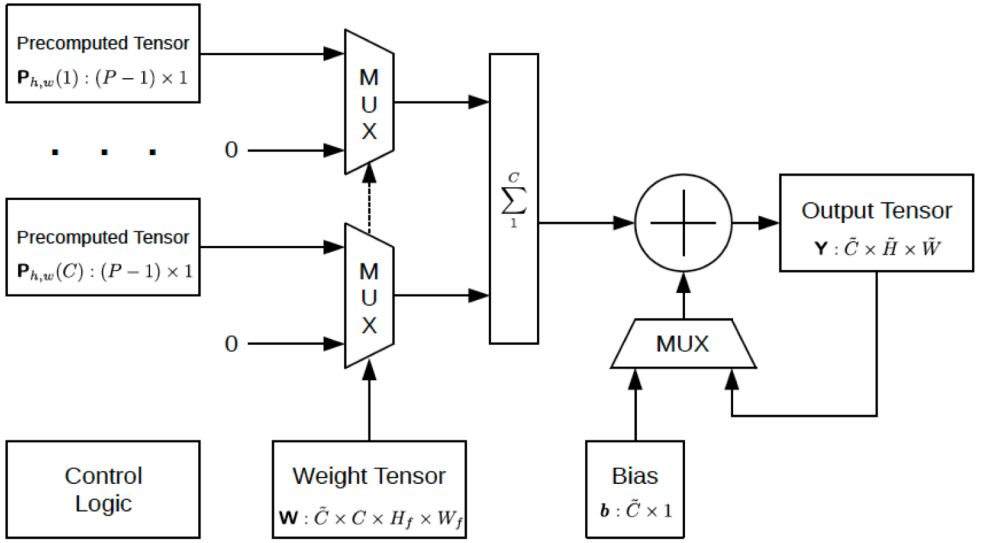

Figure 3: Array of adders.

ShiftCNN Architecture

Algorithm 2 Scheduling and control logic

for $h = 1 \ldots H$ and $w = 1 \ldots W$ **do**
    read $C \times 1$ input vector $\boldsymbol{x} = \mathbf{X}_{h,w}$
    precompute $(P-1) \times C$ matrix $\boldsymbol{P} = \mathbf{P}_{h,w}$ and write into the memory buffer
    for $\tilde{c} = 1 \ldots \tilde{C}$ **do**
        write bias $\boldsymbol{b}_{\tilde{c}}$ to the output tensor $\mathbf{Y}_{\tilde{c},h,w}$
        for $n = 1 \ldots N$ **do**
            for $h_f = 1 \ldots H_f$ and $w_f = 1 \ldots W_f$ **do**
                select value from $\{0, \boldsymbol{P}\}$ using $\mathrm{idx}_i(n)$, where $i \in \{\tilde{c}, c, h_f, w_f\}$
                add value to the output tensor $\mathbf{Y}_{\tilde{c},h,w}$

Evaluation

- ImageNet Results

Table 1: ImageNet accuracy of baseline models and corresponding ShiftCNN variants.

Network	Shifts N	Bit-width B	Top-1 Accuracy, %	Top-5 Accuracy, %	Decrease in Top-1 Accuracy, %	Decrease in Top-5 Accuracy, %
SqueezeNet	base	32	58.4	81.02		
SqueezeNet	1	4	23.01	45.93	35.39	35.09
SqueezeNet	2	4	57.39	80.31	1.01	0.71
SqueezeNet	3	4	58.39	81.01	0.01	0.01
GoogleNet	base	32	68.93	89.15		
GoogleNet	1	4	57.67	81.79	11.26	7.36
GoogleNet	2	4	68.54	88.86	0.39	0.29
GoogleNet	3	4	68.88	89.06	0.05	0.09
ResNet-18	base	32	64.78	86.13		
ResNet-18	1	4	24.61	47.02	40.17	39.11
ResNet-18	2	4	64.24(61.57)	85.79(84.08)	0.54(3.21)	0.34(2.05)
ResNet-18	3	4	64.75(64.69)	86.01(85.97)	0.03(0.09)	0.12(0.16)
ResNet-50	base	32	72.87	91.12		
ResNet-50	1	4	54.38	78.57	18.49	12.55
ResNet-50	2	4	72.20(70.38)	90.71(89.48)	0.67(2.49)	0.41(1.64)
ResNet-50	3	4	72.58(72.56)	90.97(90.96)	0.29(0.31)	0.15(0.16)

Evaluation

Table 2: High-level complexity comparison for popular CNNs

Network	Conventional Mult. Million Cycles	ShiftALU Million Cycles	Speed-Up
SqueezeNet	410	1.58	260×
GoogleNet	1750	2.55	687×
ResNet-18	1970	1.81	1090×

Table 3: FPGA utilization and power consumption estimates

ALU	LUTs	FFs	DSPs	Power, mW	Relative Power
Adders only	1644	1820	0	76	75%
ShiftCNN	4016	2219	0	102	100%
Mult. DSP	0	2048	191	423	415%
Mult. LUT	10064	4924	0	391	383%