XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks

2018-11-11

Overview

- 提出 **Binary-Weight-Networks** & **XNOR-Net**
- **Binary-Weight-Networks:** 二值化filter(weight)； smaller 32x than an equivalent network with single-precision weight values & 2 speed up
- **XNOR-Net:** 二值化input & weight； smaller 32x than an equivalent network with single-precision weight values & offering 58x speed up in CPUs

XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks

- Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, Ali Farhadi

- Torch 7

- 2016

Binary-Weight-Networks

- 将卷积操作替换为 $\mathbf{I} * \mathbf{W} \approx (\mathbf{I} \oplus \mathbf{B})\,\alpha$
- Estimating binary weights →
- Binarization in Forward & Backward

$$J(\mathbf{B}, \alpha) = \|\mathbf{W} - \alpha\mathbf{B}\|^2$$
$$\alpha^*, \mathbf{B}^* = \underset{\alpha, \mathbf{B}}{\operatorname{argmin}} J(\mathbf{B}, \alpha) \tag{2}$$

$$J(\mathbf{B}, \alpha) = \alpha^2 \mathbf{B}^\intercal \mathbf{B} - 2\alpha \mathbf{W}^\intercal \mathbf{B} + \mathbf{W}^\intercal \mathbf{W} \tag{3}$$

$$\mathbf{B}^* = \underset{\mathbf{B}}{\operatorname{argmax}} \{\mathbf{W}^\intercal \mathbf{B}\} \quad s.t. \;\; \mathbf{B} \in \{+1, -1\}^n \tag{4}$$

$$\alpha^* = \frac{\mathbf{W}^\intercal \mathbf{B}^*}{n} \tag{5}$$

By replacing $\mathbf{B}^*$ with $\operatorname{sign}(\mathbf{W})$

$$\alpha^* = \frac{\mathbf{W}^\intercal \operatorname{sign}(\mathbf{W})}{n} = \frac{\sum |\mathbf{W}_i|}{n} = \frac{1}{n}\|\mathbf{W}\|_{\ell 1} \tag{6}$$

a的最优解是W的每个元素的绝对值之和的均值

Algorithm 1 Training an L-layers CNN with binary weights:

Input: A minibatch of inputs and targets $(\mathbf{I}, \mathbf{Y})$, cost function $C(\mathbf{Y}, \hat{\mathbf{Y}})$, current weight $\mathcal{W}^t$ and current learning rate η^t.

Output: updated weight $\mathcal{W}^{t+1}$ and updated learning rate η^{t+1}.

Binarizing weight filters:
for $l = 1$ to L **do**
  for k^{th} filter in l^{th} layer **do**
    $\mathcal{A}_{lk} = \frac{1}{n}\|\mathcal{W}^t_{lk}\|_{\ell 1}$
    $\mathcal{B}_{lk} = \operatorname{sign}(\mathcal{W}^t_{lk})$
    $\widetilde{\mathcal{W}}_{lk} = \mathcal{A}_{lk}\mathcal{B}_{lk}$
$\hat{\mathbf{Y}} = \textbf{BinaryForward}(\mathbf{I}, \mathcal{B}, \mathcal{A})$ // standard forward propagation except that convolutions are computed using equation 1 or 11
$\frac{\partial C}{\partial \widetilde{\mathcal{W}}} = \textbf{BinaryBackward}(\frac{\partial C}{\partial \hat{\mathbf{Y}}}, \widetilde{\mathcal{W}})$ // standard backward propagation except that gradients are computed using $\widetilde{\mathcal{W}}$ instead of $\mathcal{W}^t$
$\mathcal{W}^{t+1} = \textbf{UpdateParameters}(\mathcal{W}^t, \frac{\partial C}{\partial \widetilde{\mathcal{W}}}, \eta_t)$ // Any update rules (*e.g.*, SGD or ADAM)
$\eta^{t+1} = \textbf{UpdateLearningrate}(\eta^t, t)$ // Any learning rate scheduling function

XNOR-Net

- Binary Dot Product(similar as Estimating binary weights)

$$\alpha^*, \mathbf{B}^*, \beta^*, \mathbf{H}* = \underset{\alpha,\mathbf{B},\beta,\mathbf{H}}{\operatorname{argmin}} \|\mathbf{X} \odot \mathbf{W} - \beta\alpha\mathbf{H} \odot \mathbf{B}\| \quad (7)$$

βH近似表示输入X

$$\gamma^*, \mathbf{C}^* = \underset{\gamma,\mathbf{C}}{\operatorname{argmin}} \|\mathbf{Y} - \gamma\mathbf{C}\| \quad (8)$$

$$\mathbf{C}^* = \operatorname{sign}(\mathbf{Y}) = \operatorname{sign}(\mathbf{X}) \odot \operatorname{sign}(\mathbf{W}) = \mathbf{H}^* \odot \mathbf{B}^* \quad (9)$$

Since $|\mathbf{X}_i|, |\mathbf{W}_i|$ are independent, knowing that $\mathbf{Y}_i = \mathbf{X}_i\mathbf{W}_i$ then, $\mathbf{E}[|\mathbf{Y}_i|] = \mathbf{E}[|\mathbf{X}_i||\mathbf{W}_i|] = \mathbf{E}[|\mathbf{X}_i|]\,\mathbf{E}[|\mathbf{W}_i|]$ therefore,

$$\gamma^* = \frac{\sum|\mathbf{Y}_i|}{n} = \frac{\sum|\mathbf{X}_i||\mathbf{W}_i|}{n} \approx \left(\frac{1}{n}\|\mathbf{X}\|_{\ell 1}\right)\left(\frac{1}{n}\|\mathbf{W}\|_{\ell 1}\right) = \beta^*\alpha^* \quad (10)$$

XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks

A typical block in CNN

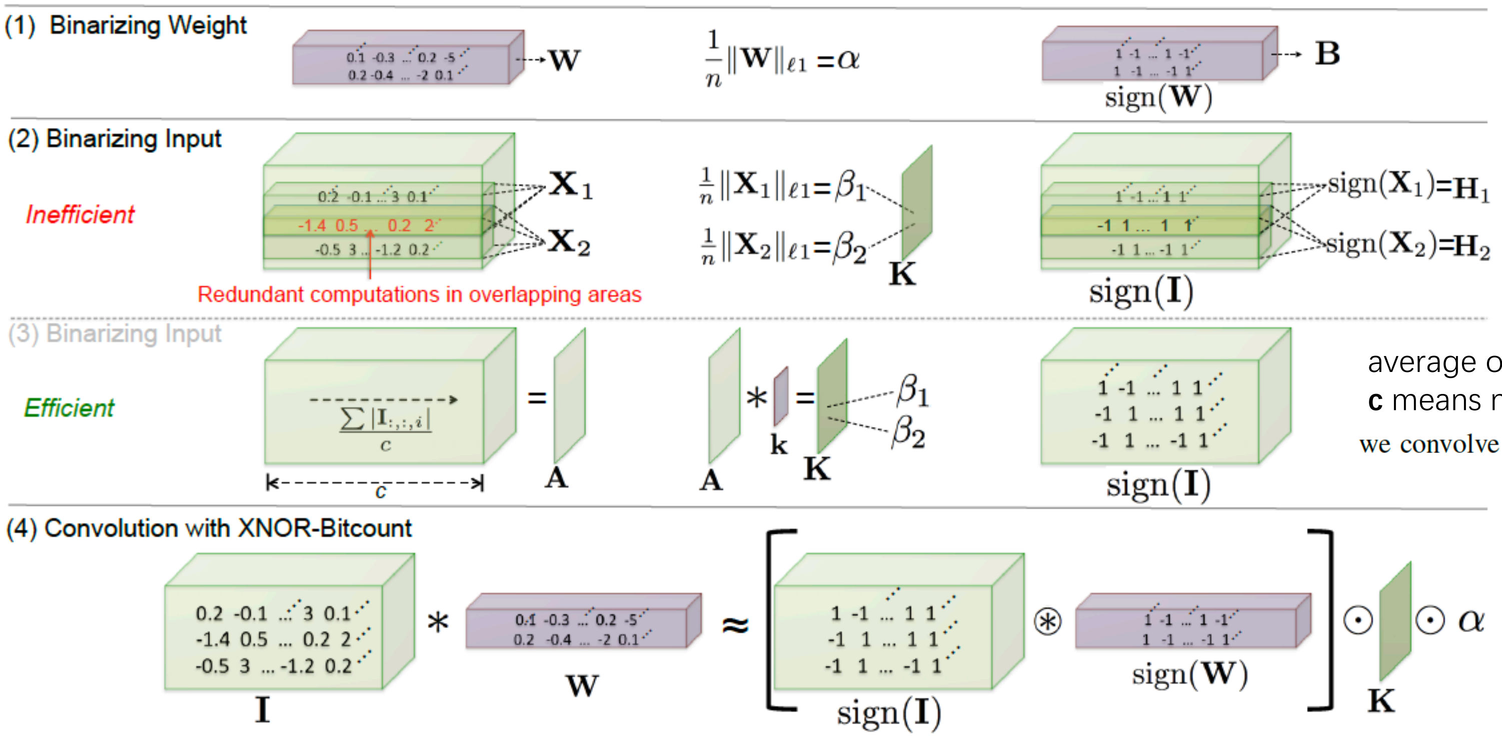
A block in XNOR-Net

- Binary Convolution（使用XNOR替代乘法提速）

$$\mathbf{I} * \mathbf{W} \approx (\operatorname{sign}(\mathbf{I}) \circledast \operatorname{sign}(\mathbf{W})) \odot \mathbf{K}\alpha \quad (11)$$

where $\circledast$ indicates a convolutional operation using XNOR and bitcount operations. This

XNOR-Net



存在重复计算

average over absolute values of the elements in the input **I**

c means number of channels

we convolve $\mathbf{A}$ with a 2D filter $\mathbf{k} \in \mathbb{R}^{w \times h}$, $\mathbf{K} = \mathbf{A} * \mathbf{k}$, where $\forall ij \;\; \mathbf{k}_{ij} = \frac{1}{w \times h}$.

Experiments

Classification Accuracy(%)									
Binary-Weight				Binary-Input-Binary-Weight				Full-Precision	
BWN		BC[11]		XNOR-Net		BNN[11]		AlexNet[1]	
Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
56.8	**79.4**	35.4	61.0	**44.2**	**69.2**	27.9	50.42	56.6	80.2

Table 1: This table compares the final accuracies (Top1 - Top5) of the full precision network with our binary precision networks; Binary-Weight-Networks(BWN) and XNOR-Networks(XNOR-Net) and the competitor methods; BinaryConnect(BC) and BinaryNet(BNN).

	ResNet-18		GoogLenet	
Network Variations	top-1	top-5	top-1	top-5
Binary-Weight-Network	60.8	83.0	65.5	86.1
XNOR-Network	51.2	73.2	N/A	N/A
Full-Precision-Network	69.3	89.2	71.3	90.0

Table 2: This table compares the final classification accuracy achieved by our binary precision networks with the full precision network in ResNet-18 and GoogLenet architectures.

精度效果

Binary-Weight-Network		
Strategy for computing α	top-1	top-5
Using equation 6	56.8	79.4
Using a separate layer	46.2	69.5

(a)

XNOR-Network		
Block Structure	top-1	top-5
C-B-A-P	30.3	57.5
B-A-C-P	44.2	69.2

(b)

Table 3: In this table, we evaluate two key elements of our approach; computing the optimal scaling factors and specifying the right order for layers in a block of CNN with binary input. (a) demonstrates the importance of the scaling factor in training binary-weight-networks and (b) shows that our way of ordering the layers in a block of CNN is crucial for training XNOR-Networks. C,B,A,P stands for Convolutional, BatchNormalization, Acive function (here binary activation), and Pooling respectively.

- **Binary-Weight-Networks:** 二值化filter(weight)； smaller 32x than an equivalent network with single-precision weight values & 2 speed up

- **XNOR-Net:** 二值化input & weight； smaller 32x than an equivalent network with single-precision weight values & offering 58x speed up in CPUs

压缩效果