XNOR-NET: IMAGENET CLASSIFICATION USING BINARY CONVOLUTIONAL NEURAL NETWORKS

RYAN MCCAMPBELL

2019 SPRING @ HTTPS://QDATA.GITHUB.IO/DEEP2READ

PROBLEM

- Ordinary neural networks are expensive to store and evaluate
- Not good for embedded and mobile platforms
- How to make them more efficient?

APPROACHES

- More compact networks
- Compressing pre-trained networks
- Quantizing parameters
- Binarizing parameters/activations

BINARY-WEIGHT-NETWORK

- Idea: store weights as binary vectors, ± 1, plus scale

$\boldsymbol{W} \approx \alpha \boldsymbol{B}$, $\alpha > 0$, $\boldsymbol{B} \in \{-1,1\}↑n$

$\boldsymbol{I} * \boldsymbol{W} \approx (\boldsymbol{I} \oplus \boldsymbol{B}) \alpha$

- $\oplus$: convolution using only addition/subtraction
- Eliminates most multiplications
- ~32x storage reduction

BINARY-WEIGHT-NETWORK

- Minimize $J(B,\alpha)=\|W-\alpha B\|^2$

$B^* = sign(W)$

$\alpha^* = \sum |W_i| / n = 1/n \, \|W\|_1$

TRAINING

- Binarize weights during forward pass and backwards pass
- Use full-parameter weights for update
- Approximate $\partial / \partial x \, sign(x) = 1[|r| \leq 1]$

XNOR-NET

- Binarize both weights and inputs
- Convolutions using efficient binary operations: shift, XNOR & bit-count

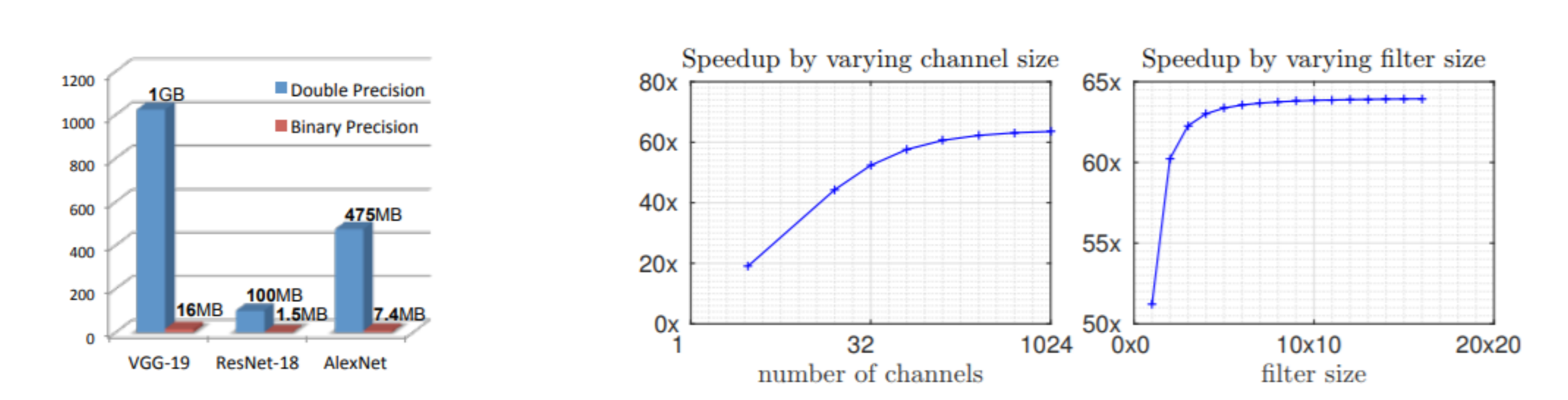
$$\mathbf{I} * \mathbf{W} \approx (\text{sign}(\mathbf{I}) \circledast \text{sign}(\mathbf{W})) \odot \mathbf{K}\alpha$$

- $K_{ij} = \beta = 1/n \, \|subtensor\ of\ X\ at\ ij\|_1$
- $K = A * k,\ A = \sum |I_{:,:,i}|/c,\ k_{ij} = 1/wh$

TRAINING XNOR-NET

- Batch normalization first
- Binary activation: compute K and sign(I)
- Binary convolution
- Pool after convolution

RESULTS



1200
1000
800
600
400
200
0
1GB
16MB
100MB
1.5MB
475MB
7.4MB
Double Precision
Binary Precision
VGG-19
ResNet-18
AlexNet
Speedup by varying channel size
80x
60x
40x
20x
0x
1
32
1024
number of channels
Speedup by varying filter size
65x
60x
55x
50x
0x0
10x10
20x20
filter size

RESULTS

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