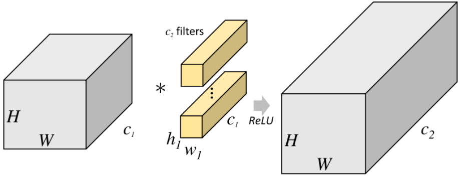
Mobile Network

Chen Shangyu

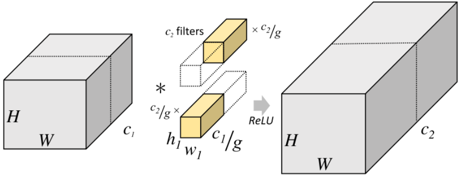
October 20, 2018

Outline

Group Convolution



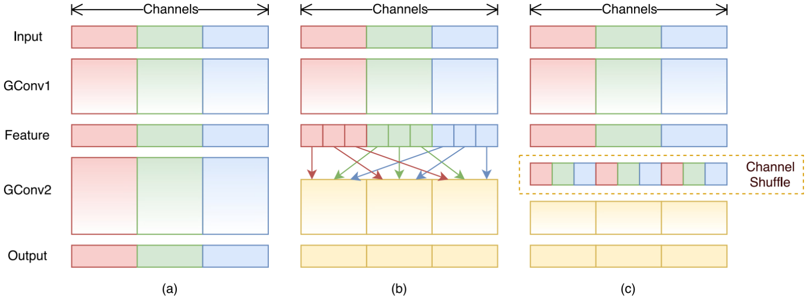
(a) Convolution.



(b) Convolution with filter groups.

- Compression ratio / Computation reduced: $\frac{1}{g}$

ShuffleNet


Channels
Channels
Channels
Input
GConv1
Feature
GConv2
Output
Channel Shuffle
(a)
(b)
(c)

IGVC1: Interleaved Group Convolutions for Deep Neural Networks

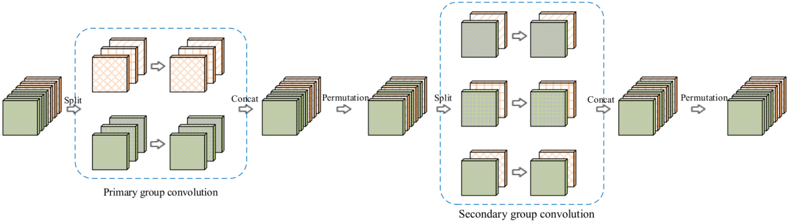


Figure: Illustrating the interleaved group convolution, with $L = 2$ primary partitions and $M = 3$ secondary partitions. The convolution for each primary partition in primary group convolution is spatial. The convolution for each secondary partition in secondary group convolution is point-wise (1×1).

IGCV2: Interleaved Structured Sparse Convolutional Neural Networks

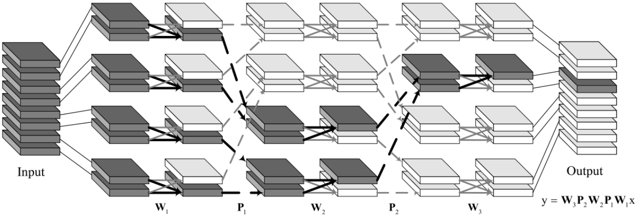


Figure: IGCV2: the Interleaved Structured Sparse Convolution. $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{W}_3$ (denoted as solid arrows) are sparse block matrices corresponding to group convolutions. $\mathbf{P}_1$ and $\mathbf{P}_1$ (denoted as dashed arrows) are permutation matrices. The resulting composed kernel $\mathbf{W}_3\mathbf{P}_2\mathbf{W}_2\mathbf{P}_1\mathbf{W}_1$ is ensured to satisfy the complementary condition which guarantees that **for each output channel, there exists one and only one path connecting the output channel to each input channel**. The bold line connecting gray feature maps shows such a path.

IGCV3: Interleaved Low-Rank Group Convolutions for Efficient Deep Neural Networks

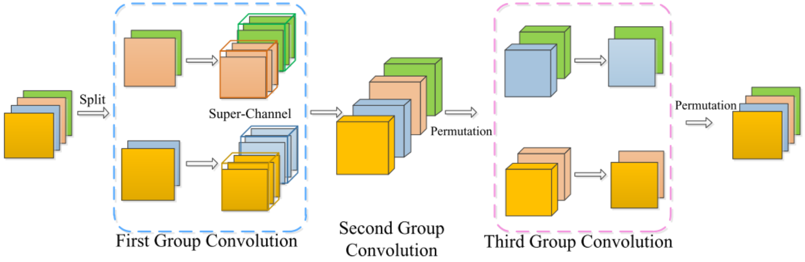


Figure: Illustrating the interleaved branches in IGCV3 block. The first group convolution is a group 1×1 convolution with $G_1 = 2$ groups. The second is a channel-wise spatial convolution. The third is a group 1×1 convolution with $G_2 = 2$ groups.

IGCV Summary

- IGCV2 extends IGCV1 by decomposing the convolution matrix into more structured sparse matrices.
- IGCV3 extends IGCV2 by using **low-rank** group convolutions to replace group convolution.

ShuffleNet V2

- This work proposes to evaluate the **direct metric** on the target platform, beyond only considering FLOPs.
- The discrepancy between the indirect (FLOPs) and direct (speed) metrics can be attributed to two main reasons: 1) Memory Access Cost (MAC) 2) Degree of Parallelism.
- Four guidelines:
  - Equal channel width minimizes memory access cost.
  - Excessive group convolution increases MAC.
  - Network fragmentation reduces degree of parallelism:

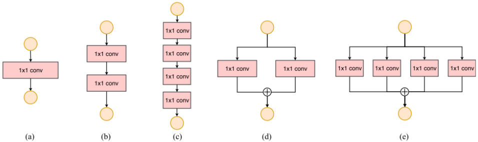


Figure: Building blocks used in experiments for guideline 3. (a) 1-fragment. (b) 2-fragment-series. (c) 4-fragment-series. (d) 2-fragment-parallel. (e) 4-fragment-parallel.

ShuffleNet V2 (Cont.)

- Element-wise (ReLU, AddTensor, AddBias) operations are non-negligible: Therefore it use concatenate.

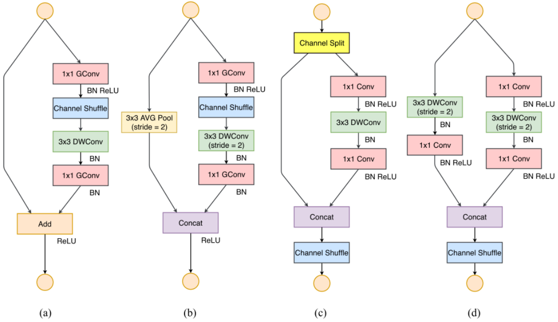
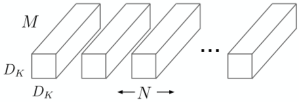
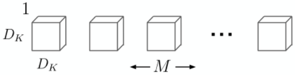


Figure: Building blocks of ShuffleNet and this work. (a): the basic ShuffleNet unit; (b) the ShuffleNet unit for spatial down sampling (2 $\times$); (c) our basic unit; (d) our unit for spatial down sampling (2). **DWConv**: depthwise convolution. **GConv**: group convolution.

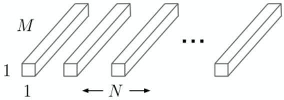
Depth-Wise Convolution (MobileNet V1)



(a) Standard Convolution Filters



(b) Depthwise Convolutional Filters



(c) 1×1 Convolutional Filters called Pointwise Convolution in the context of Depthwise Separable Convolution

Method	#. Parameter	#.Computation
Standard	$H \times W \times M \times N$	$H^2 \times W^2 \times M \times N$
Depth-Wise	$H \times W \times M + N$	$H^2 \times W^2 \times M + M \times N \times H \times W$

Linear Bottlenecks

- Definition: A convolution layer without ReLU.
- Motivation:
  - ReLU acts as linear transformer if input is non-negative.
  - ReLU is capable of preserving complete information about the input manifold, but only if the input manifold lies in a low-dimensional subspace of the input space.

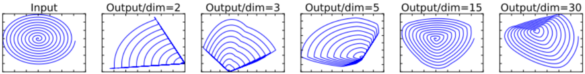


Figure: Examples of ReLU transformations of low-dimensional manifolds embedded in higher-dimensional spaces. In these examples the initial spiral is embedded into an n-dimensional space using random matrix T followed by ReLU, and then projected back to the 2D space using $T1$. In examples above $n = 2, 3$ result in information loss where certain points of the manifold collapse into each other, while for $n = 15$ to 30 the transformation is highly non-convex

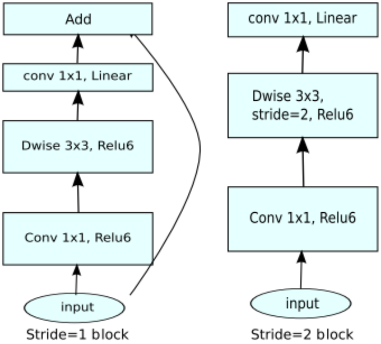
Inverted Residuals

- Traditionally, number of channels in a bottlenet unit: a, b, c shows: $a < b$ and $c < b$. First and last layer is 1×1, middle is 3×3.

Input	Operator	Output
$h \times w \times k$	1x1 conv2d , ReLU6	$h \times w \times (tk)$
$h \times w \times tk$	3x3 dwise s=s, ReLU6	$\frac{h}{s} \times \frac{w}{s} \times (tk)$
$\frac{h}{s} \times \frac{w}{s} \times tk$	linear 1x1 conv2d	$\frac{h}{s} \times \frac{w}{s} \times k'$

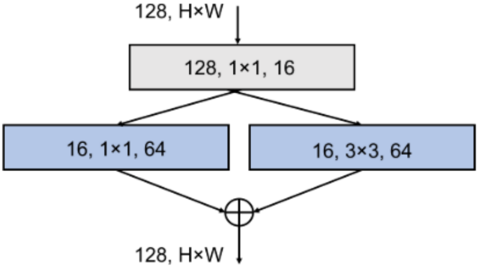
Figure: Inverted bottleneck residual block transforming from k to $k^{'}$ channels, with stride s, and expansion factor t

MobileNet V2



SqueezeNet

- Squeeze layer: 1×1 convolution to decrease channels
- Expand layer

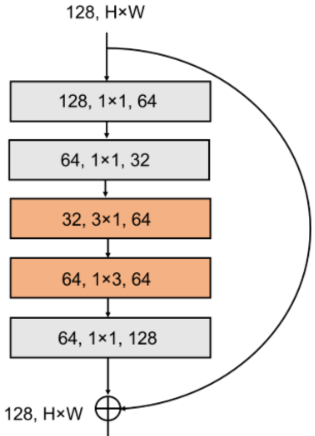


Strategies Behind

- Replace 3×3 filters with 1×1 filter: use more 1×1 filters (9 $\times$ fewer parameters / Computation)
- Decrease the number of input channels to 3×3 filter.
- Downsample late

SqueezeNext

- Two-stage bottleneck module to reduce the number ofinput channels.
- Low rank separable convolutions to replace 3×3 filter.



Summary

- Group convolution.
- Group communication: Point-Wise Convolution, Channel Shuffle, Interleaved Group Convolution.
- Inverted bottleneck.