

# ResNeXt

Aggregated Residual Transformations for Deep  
Neural Networks

허진규, 전태호

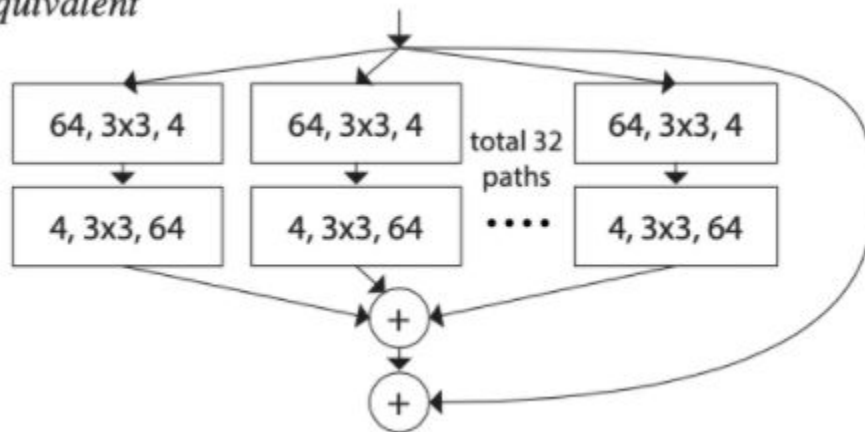
# Abstract

Cardinality 기법을 통해서 Resnet의 성능을 높이려고 했다.

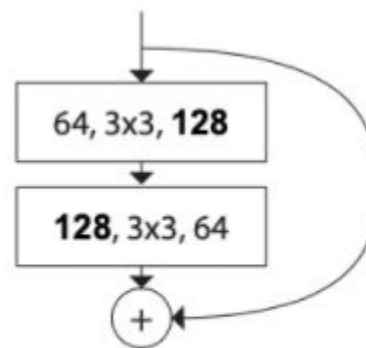
ILSVRC2016 에서 2등을 했다.

# Cardinality

*equivalent*



128(4\*32)->64 layer

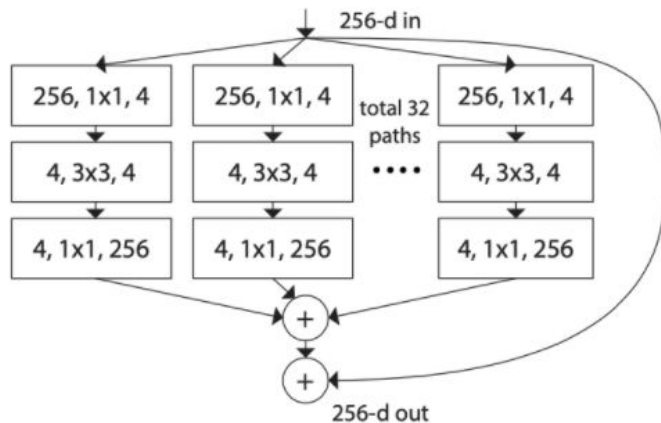
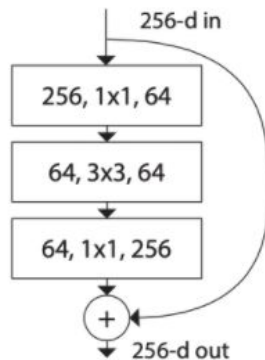


128->64 layer

total channel = separated channel \* cardinality

# Introduction

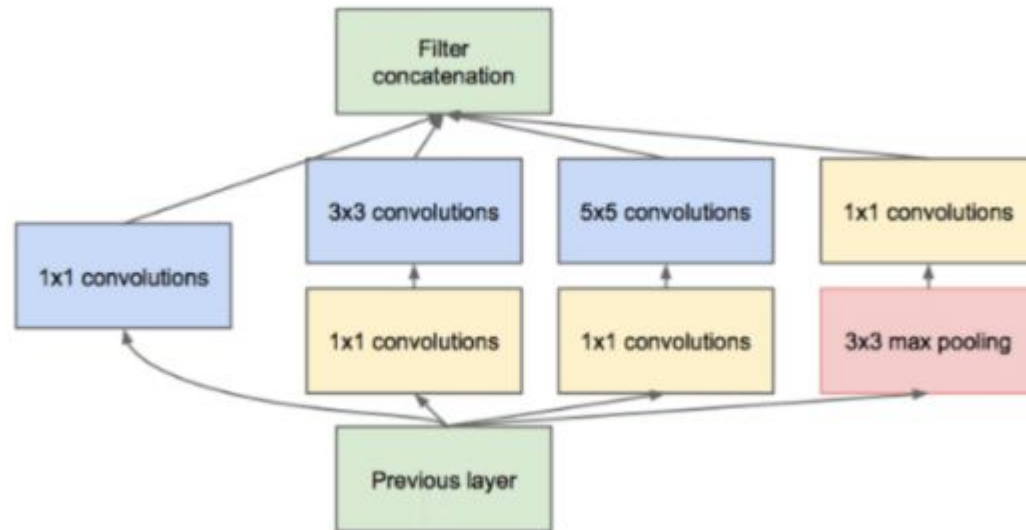
- 깊어지면 param의 수가 늘어나서 어려움
- Inception에서 split-transform-merge strategy를 이용해서 적은 연산량과 높은 정확도를 이끔
- 1x1 conv로 transform, concat으로 merge하며 적용시킴

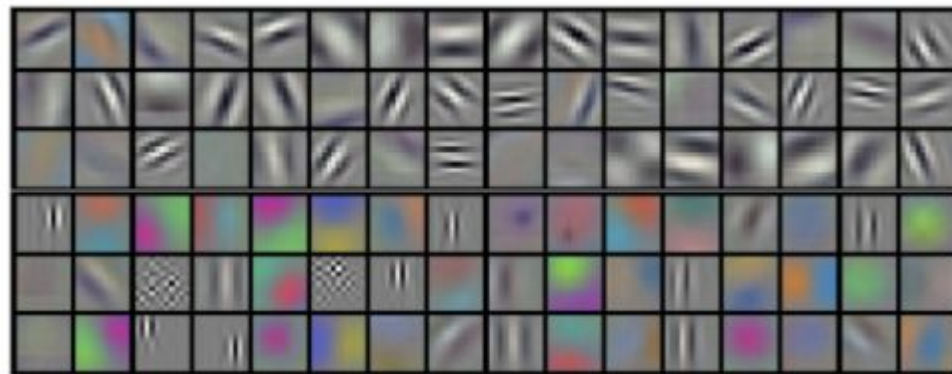
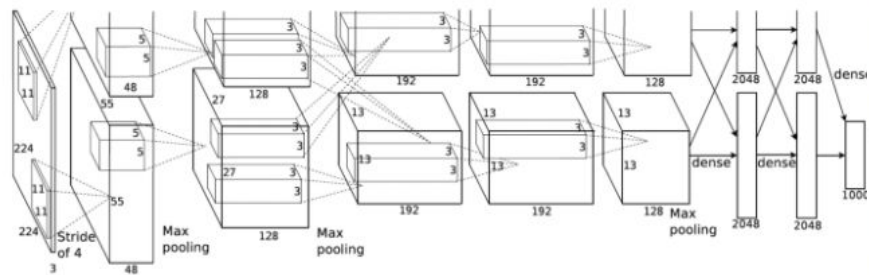


<ResNeXt의 블록 구조>

# Related Work

1. Multi-branch convolutional networks (inception model)
2. Grouped convolutions (2 gpu)
3. Compressing convolutional networks
4. Ensembling (실 패(?))





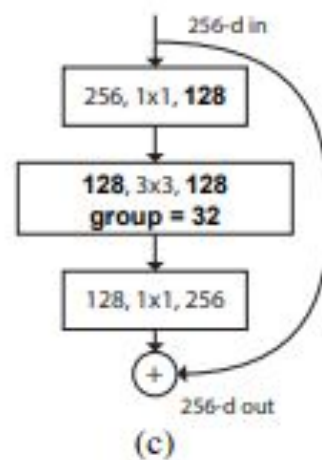
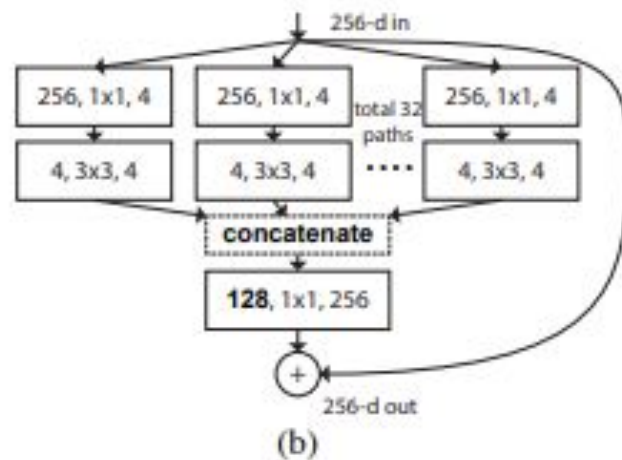
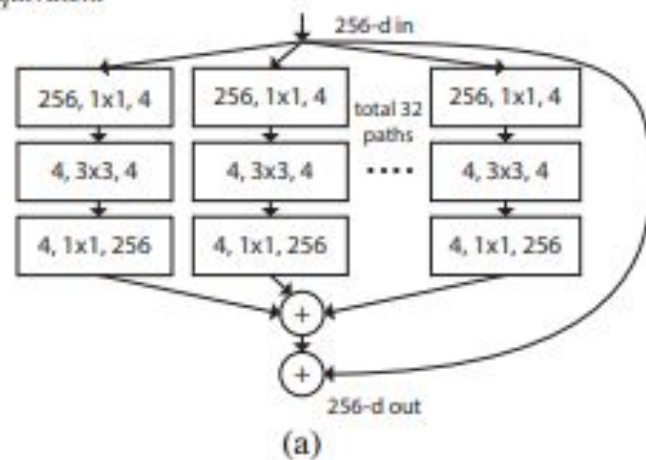
# Method

- conv를 통과할때마다 output 반으로 감소
- ResNet보다 채널 수는 더 많지만 연산 수는 더 적은 것을 볼 수 있다.
- ResNeXt는 각 블록이 3개의 conv를 사용해야 성능 향상에 도움이 된다.

stage	output	ResNet-50	ResNeXt-50 (32×4d)
conv1	112×112	7×7, 64, stride 2	7×7, 64, stride 2
conv2	56×56	3×3 max pool, stride 2	3×3 max pool, stride 2
		$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128, C=32 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3	28×28	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256, C=32 \\ 1 \times 1, 512 \end{bmatrix} \times 4$
conv4	14×14	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512, C=32 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$
conv5	7×7	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 1024 \\ 3 \times 3, 1024, C=32 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	global average pool 1000-d fc, softmax	global average pool 1000-d fc, softmax
# params.		<b>25.5</b> ×10 <sup>6</sup>	<b>25.0</b> ×10 <sup>6</sup>
FLOPs		<b>4.1</b> ×10 <sup>9</sup>	<b>4.2</b> ×10 <sup>9</sup>



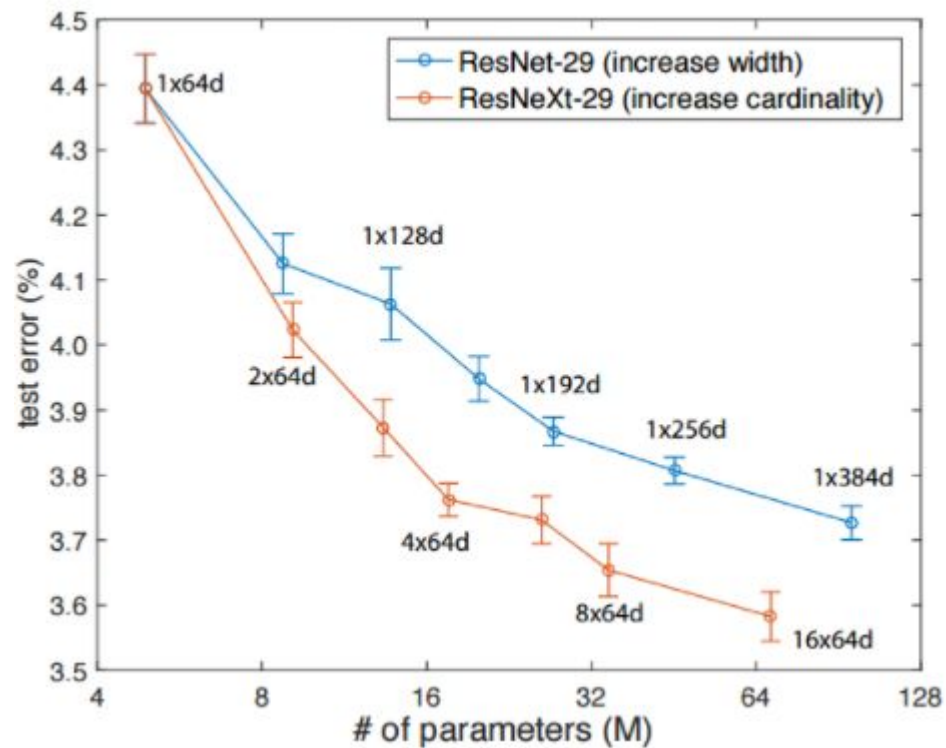
*equivalent*



# Cardinality vs Width

- Width보다 Cardinality를 증가시키는 것이 더 효율적이다.
- 효율적이라는 말은 동일한 파라미터 수 대비 더 좋은 성능을 의미한다.

	setting	top-1 error (%)
ResNet-50	$1 \times 64d$	23.9
ResNeXt-50	$2 \times 40d$	23.0
ResNeXt-50	$4 \times 24d$	22.6
ResNeXt-50	$8 \times 14d$	22.3
ResNeXt-50	$32 \times 4d$	<b>22.2</b>
ResNet-101	$1 \times 64d$	22.0
ResNeXt-101	$2 \times 40d$	21.7
ResNeXt-101	$4 \times 24d$	21.4
ResNeXt-101	$8 \times 14d$	21.3
ResNeXt-101	$32 \times 4d$	<b>21.2</b>



# Cardinality vs Deep vs Width

- ResNeXt-101 >  
ResNet-101 wider >  
ResNet-200

- 즉,

cardinality > width > deep

	setting	top-1 err (%)	top-5 err (%)
<i>1× complexity references:</i>			
ResNet-101	1 × 64d	22.0	6.0
ResNeXt-101	32 × 4d	21.2	5.6
<i>2× complexity models follow:</i>			
ResNet- <b>200</b> [15]	1 × 64d	21.7	5.8
ResNet-101, wider	1 × <b>100d</b>	21.3	5.7
ResNeXt-101	<b>2</b> × 64d	20.7	5.5
ResNeXt-101	<b>64</b> × 4d	<b>20.4</b>	<b>5.3</b>

# ResNeXt Architecture

stage	output	ResNet-50	ResNeXt-50 (32×4d)
conv1	112×112	7×7, 64, stride 2	7×7, 64, stride 2
conv2	56×56	3×3 max pool, stride 2	3×3 max pool, stride 2
		$\begin{bmatrix} 1\times 1, 64 \\ 3\times 3, 64 \\ 1\times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times 1, 128 \\ 3\times 3, 128, C=32 \\ 1\times 1, 256 \end{bmatrix} \times 3$
conv3	28×28	$\begin{bmatrix} 1\times 1, 128 \\ 3\times 3, 128 \\ 1\times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times 1, 256 \\ 3\times 3, 256, C=32 \\ 1\times 1, 512 \end{bmatrix} \times 4$
conv4	14×14	$\begin{bmatrix} 1\times 1, 256 \\ 3\times 3, 256 \\ 1\times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times 1, 512 \\ 3\times 3, 512, C=32 \\ 1\times 1, 1024 \end{bmatrix} \times 6$
conv5	7×7	$\begin{bmatrix} 1\times 1, 512 \\ 3\times 3, 512 \\ 1\times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times 1, 1024 \\ 3\times 3, 1024, C=32 \\ 1\times 1, 2048 \end{bmatrix} \times 3$
	1×1	global average pool 1000-d fc, softmax	global average pool 1000-d fc, softmax
# params.		<b>25.5</b> ×10 <sup>6</sup>	<b>25.0</b> ×10 <sup>6</sup>
FLOPs		<b>4.1</b> ×10 <sup>9</sup>	<b>4.2</b> ×10 <sup>9</sup>

Thanks for listening