CVPR 2017
Aggregated Residual Transformations for Deep Neural Networks

2022.07.27

논문 리뷰

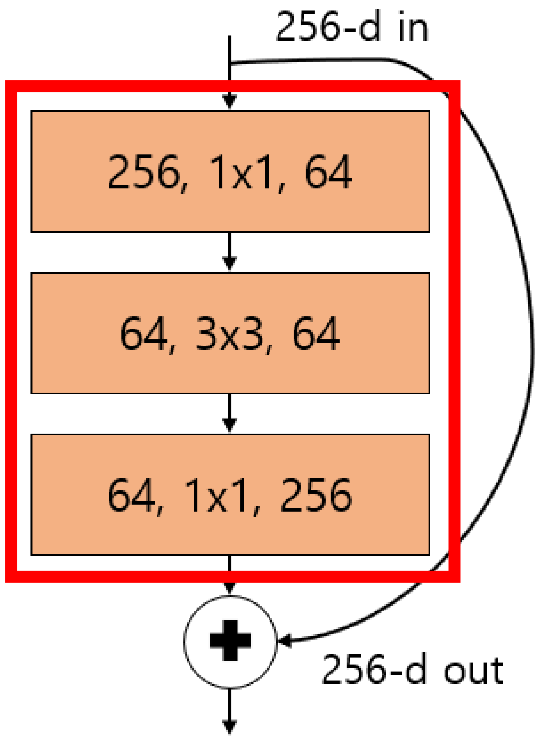
배성훈

Aggregated Residual Transformations for Deep Neural Networks (CVPR 2017)

- **Research Background:**
  - 문제: Network Depth가 깊어질수록 **Hyper-parameters 증가**하는 어려움 발생
  - 해결: **Cardinality** 라는 새로운 차원의 도입과 **Hyper-parameter 효율적 조절**을 통해 성능 향상 (ResNet에서 발전)

Repeating layer 전략(VGG, ResNet) + "Cardinarity" + Split/Transform/Merge 전략(InceptionNet) + Grouped convolution

Repeating layer

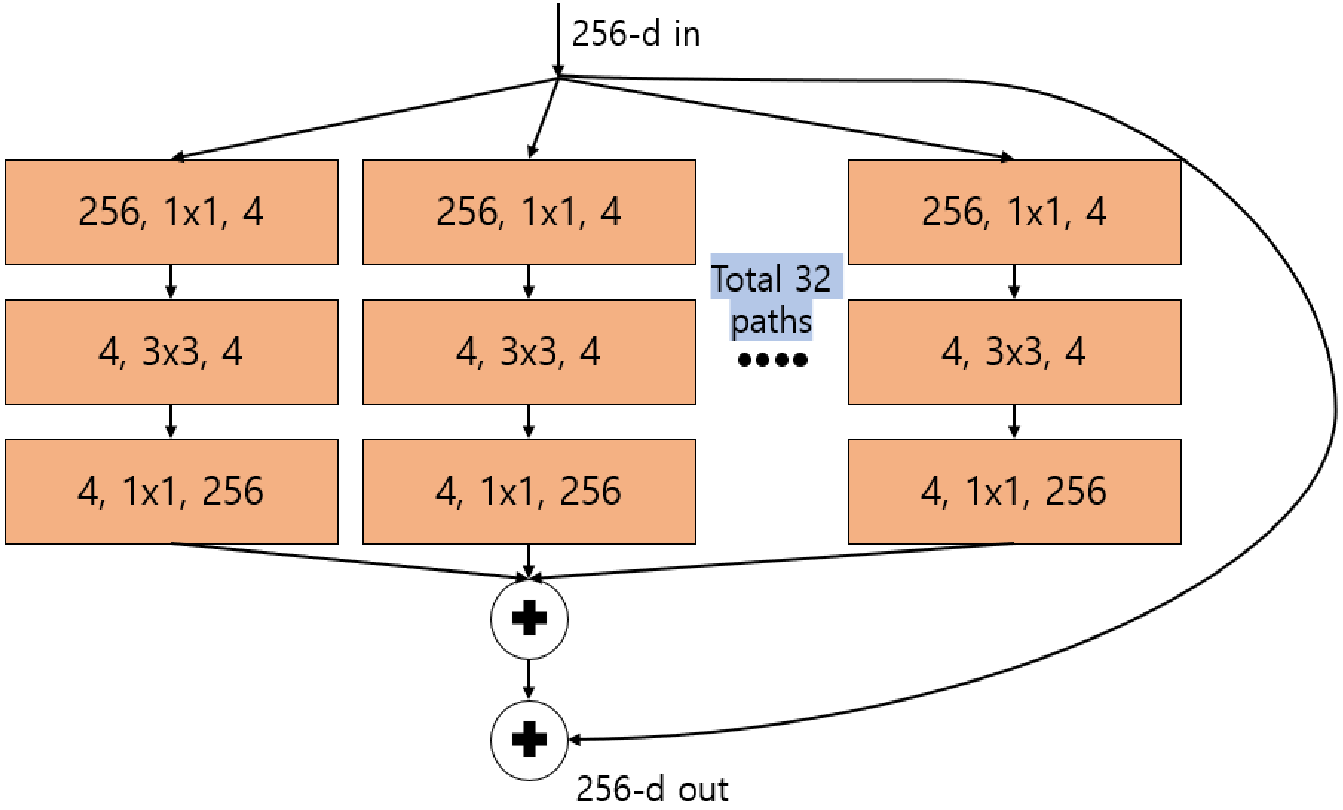


ResNet

Aggregated Residual Transformations for Deep Neural Networks (CVPR 2017)

- **Research Background:**

Repeating layer 전략(VGG, ResNet) + “**Cardinarity**” + Split/Transform/Merge 전략(InceptionNet) + Grouped convolution



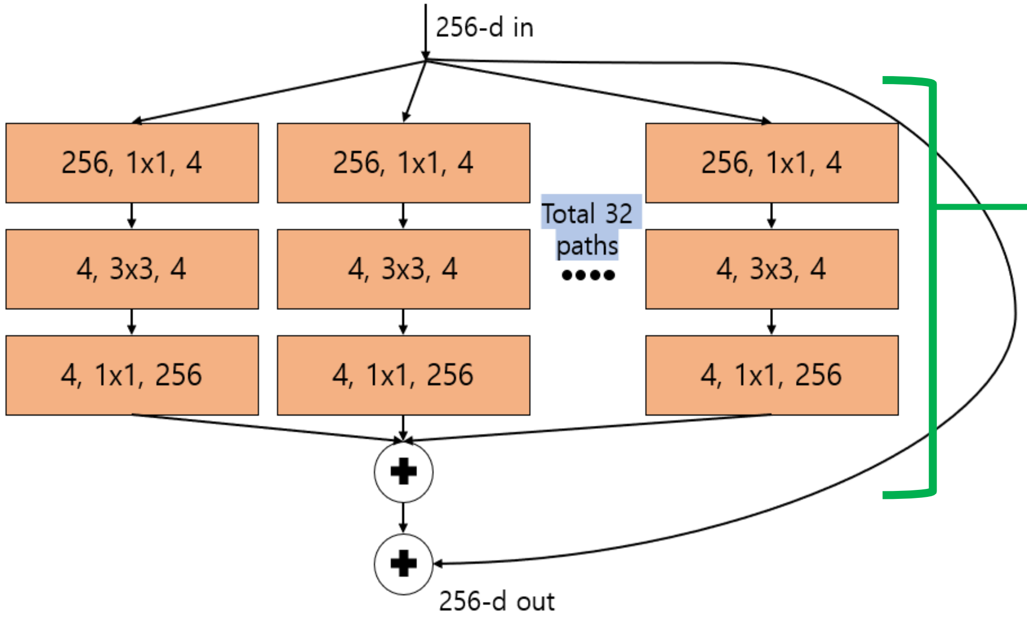
ResNeXt (C=32, d=4)

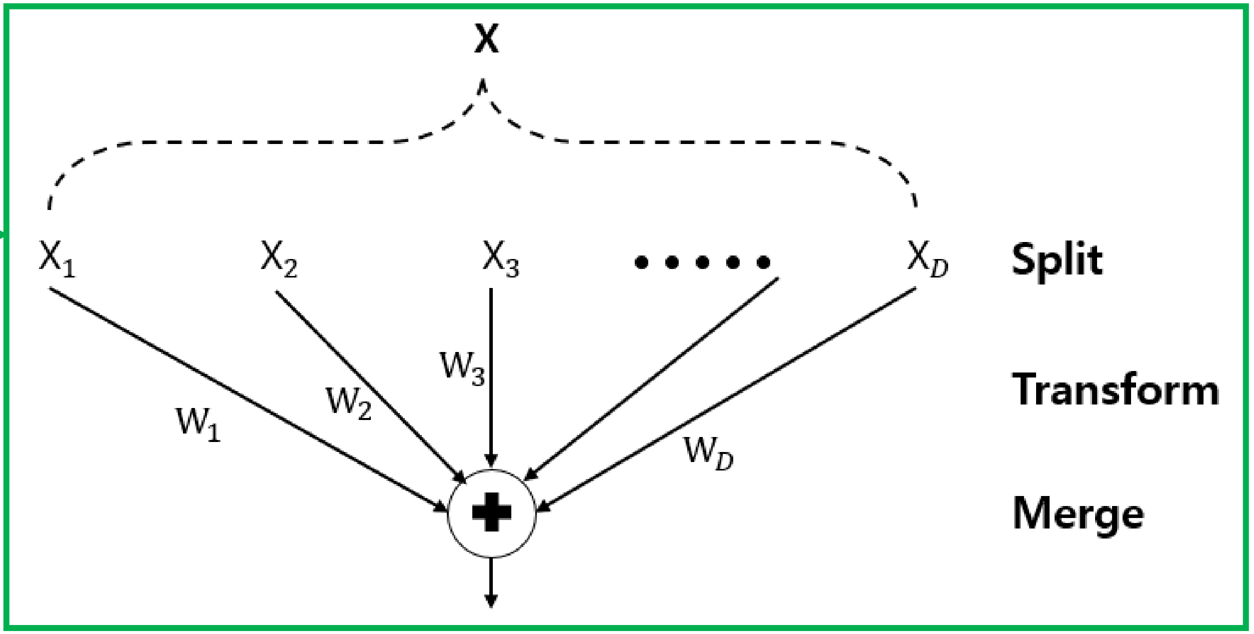
C: **Cardinality**, ConvNet을 몇 개로 나눌지
d: ConvNet의 Channel

Aggregated Residual Transformations for Deep Neural Networks (CVPR 2017)

- **Research Background:**

Repeating layer 전략(VGG, ResNet) + “Cardinarity” + **Split/Transform/Merge 전략**(InceptionNet) + Grouped convolution





Split: x를 split $(x_1, x_2, x_3, \ldots, x_D)$
Transform: weigh을 곱함 $(x_1W_1, x_2W_2, \ldots, x_DW_D)$
Merge: 전체 결과 합 $(x_1W_1 + x_2W_2 + \cdots + x_DW_D)$

$$F(x) = \boxed{\sum_{i=1}^{C} T_i(x)} + x_{identity} \rightarrow y = x + \boxed{\sum_{i=1}^{C} T_i(x)}$$

Aggregated Residual Transformations for Deep Neural Networks (CVPR 2017)

- **Research Background:**
  - **<u>Grouped convolution</u>** 사용으로 구현을 간단하게 함

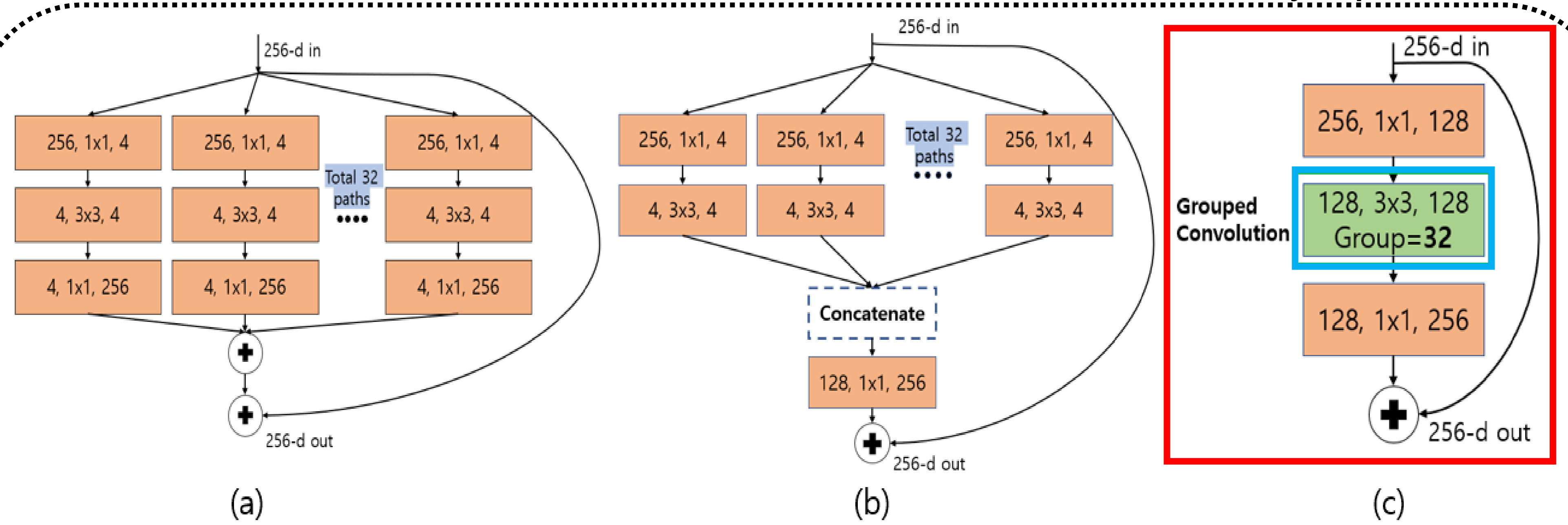
Repeating layer 전략(VGG, ResNet) + “Cardinarity” + Split/Transform/Merge 전략(InceptionNet) + **<u>Grouped convolution</u>**

Relations between cardinality(C) and width(d)

가장 최적의 성능

cardinality C	1	2	4	8	32
width of bottleneck d	64	40	24	14	4
width of group conv.	64	80	96	112	128

Fast, Easy Implementation

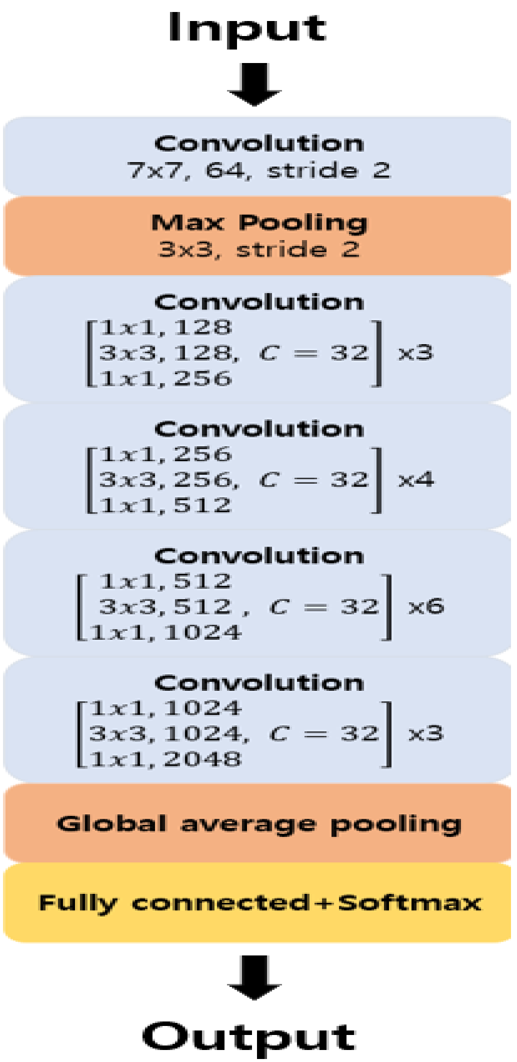


Equivalent building blocks of ResNeXt
Performance: (a) = (b) = (c)

Aggregated Residual Transformations for Deep Neural Networks (CVPR 2017)

- **Method:**
  - 같은 Spatial map인 경우, block들은 같은 hyper-parameter 공유 **=> <u>Hyper-Parameter 선택 부담감↓</u>**

ResNeXt Architecture

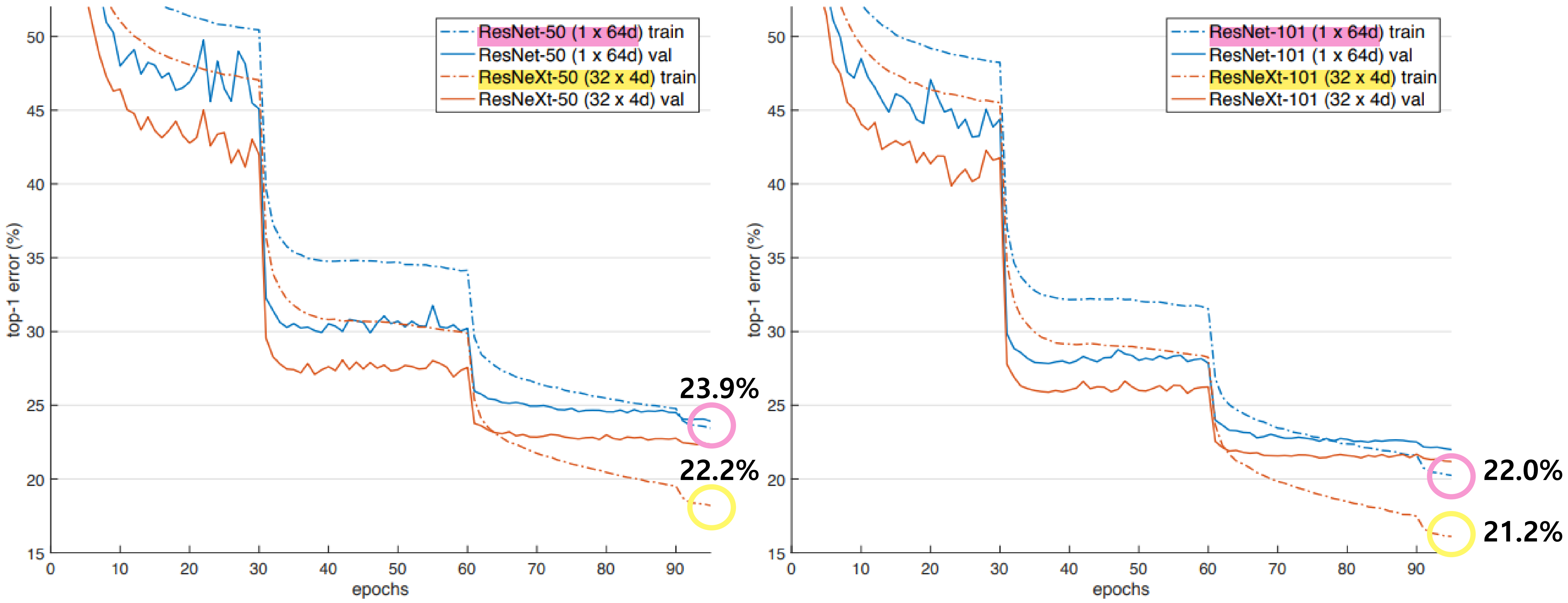


Aggregated Residual Transformations for Deep Neural Networks (CVPR 2017)

- **Experiment:**

  <u>Cardinality</u>라는 새로운 차원을 활용한 ResNeXt가 기존의 ResNet 보다 더 **<u>좋은 성능을 달성</u>**

Training curves on ImageNet-1K



Aggregated Residual Transformations for Deep Neural Networks (CVPR 2017)

- **Experiment:**
  1. ResNet 보다 ResNeXt가 더 좋은 성능을 보임
  2. Cardinality(C)와 width(d)의 설정 값에 따른 비교, 32 x 4d가 가장 좋은 성능을 보임

Ablation experiments on ImageNet-1K

	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	$32 \times 4d$	**22.2**
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	$32 \times 4d$	**21.2**

Aggregated Residual Transformations for Deep Neural Networks (CVPR 2017)

- **Experiment:**
  - Cardinality의 증가가 Network의 wider, deeper보다 더 좋은 성능을 보임

Increasing Cardinality vs Deeper/Wider

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

성능: deeper < wider << **Cardinality**

State-of-the-art models on ImageNet-1K Validation

		224×224		320×320 / 299×299	
		top-1 err	top-5 err	top-1 err	top-5 err
Depth	ResNet-101 [14]	22.0	6.0	-	-
	ResNet-200 [15]	21.7	5.8	20.1	4.8
Width	Inception-v3 [39]	-	-	21.2	5.6
	Inception-v4 [37]	-	-	20.0	5.0
	Inception-ResNet-v2 [37]	-	-	19.9	4.9
Cardinality	ResNeXt-101 (**64** × **4d**)	20.4	5.3	**19.1**	**4.4**

한줄평: 개인적으로, 본 논문은 Network Depth와 Width를 함께 활용한 방법이라 생각함
Cardinality는 결국 convolution을 얼마나 분할할지 나타내는 것 **->** Residual block을 wider하게 만듬