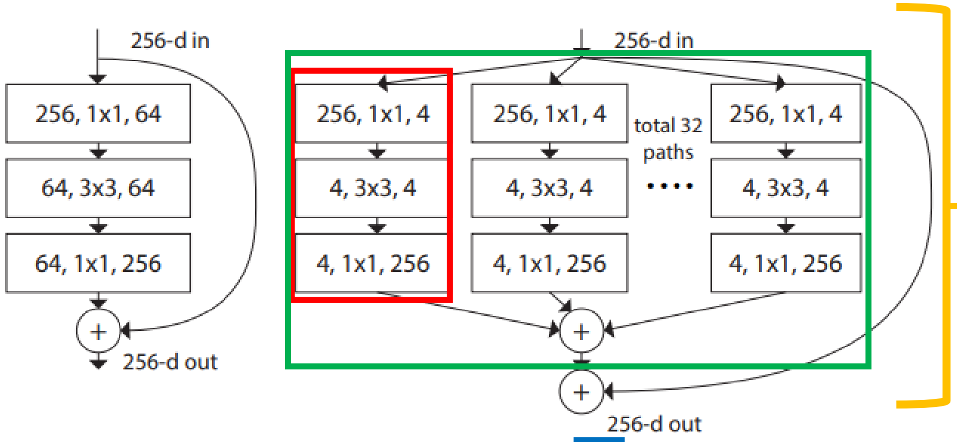
Aggregated Residual Transformations for Deep Neural Networks (CVPR 2017)

- **연구동기 & Method :**

  문제: Network depth ↑ -> Hyper-parameters 선택 양 ↑

  해결: Hyper-parameter 효율적 조절 + 계산 복잡도 유지 or 감소 + 성능 향상 (ResNet에서 발전).

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

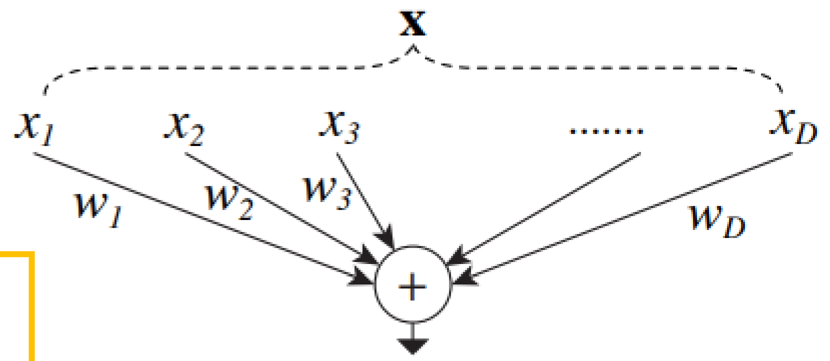


ResNet

ResNeXt (C=32, d=4)

C: cardinality, ConvNet을 몇 개로 나눌지

d: ConvNet의 Channel



Split/Transform/Merge

Complexity ↓

Performance ↑

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

$$y = x + \sum_{i=1}^{C} T_i(x)$$

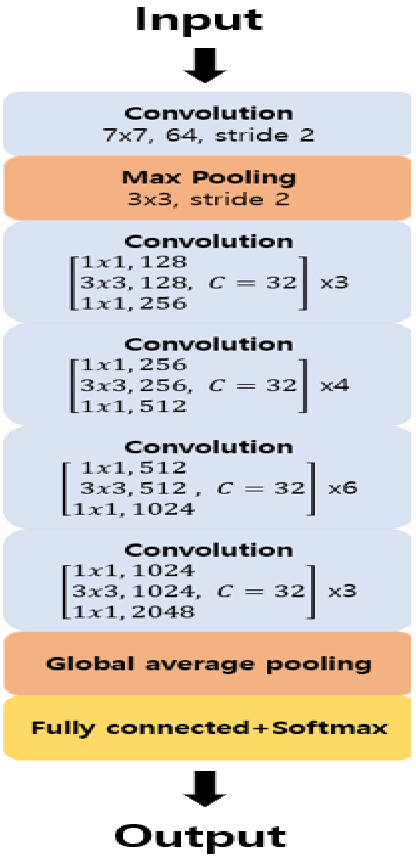
Split: x를 split $(x_1, x_2, x_3, \dots, x_D)$

Transform: weigh을 곱함 $(x_1W_1, x_2W_2, \dots, x_DW_D)$

Merge: 전체 결과 합 $(x_1W_1 + x_2W_2 + \cdots + x_DW_D)$

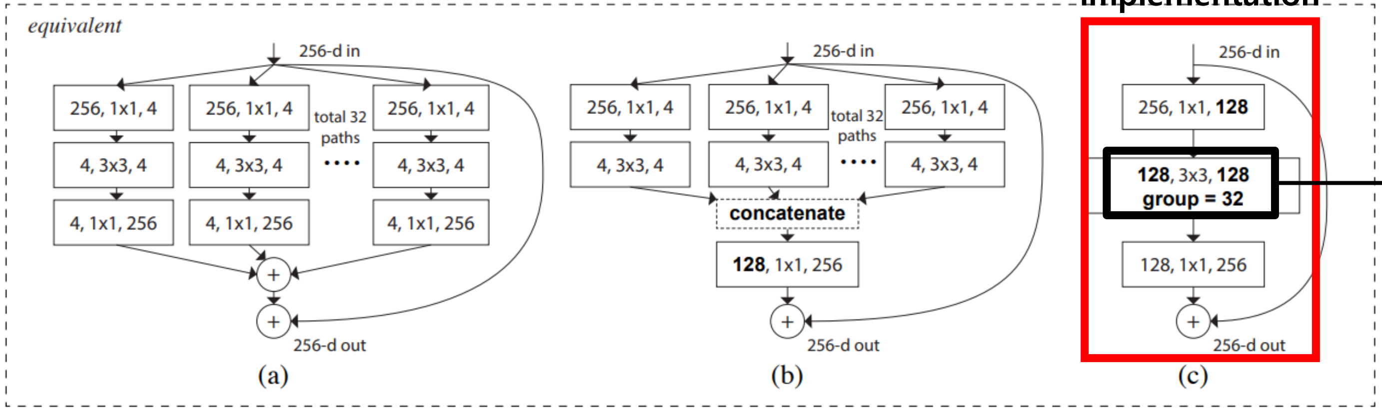
Aggregated Residual Transformations for Deep Neural Networks (CVPR 2017)

- **Method:**
  같은 spatial map인 경우, block들은 같은 hyperparameter 공유 => hyper-parameter 선택 부담감↓
  Grouped convolution 의 사용으로 구현을 간단하게!



ResNeXt Architecure

Fast, Easing implementation



Grouped Convolution

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

Relations between cardinality(C) and width(d)

가장 최적의 성능

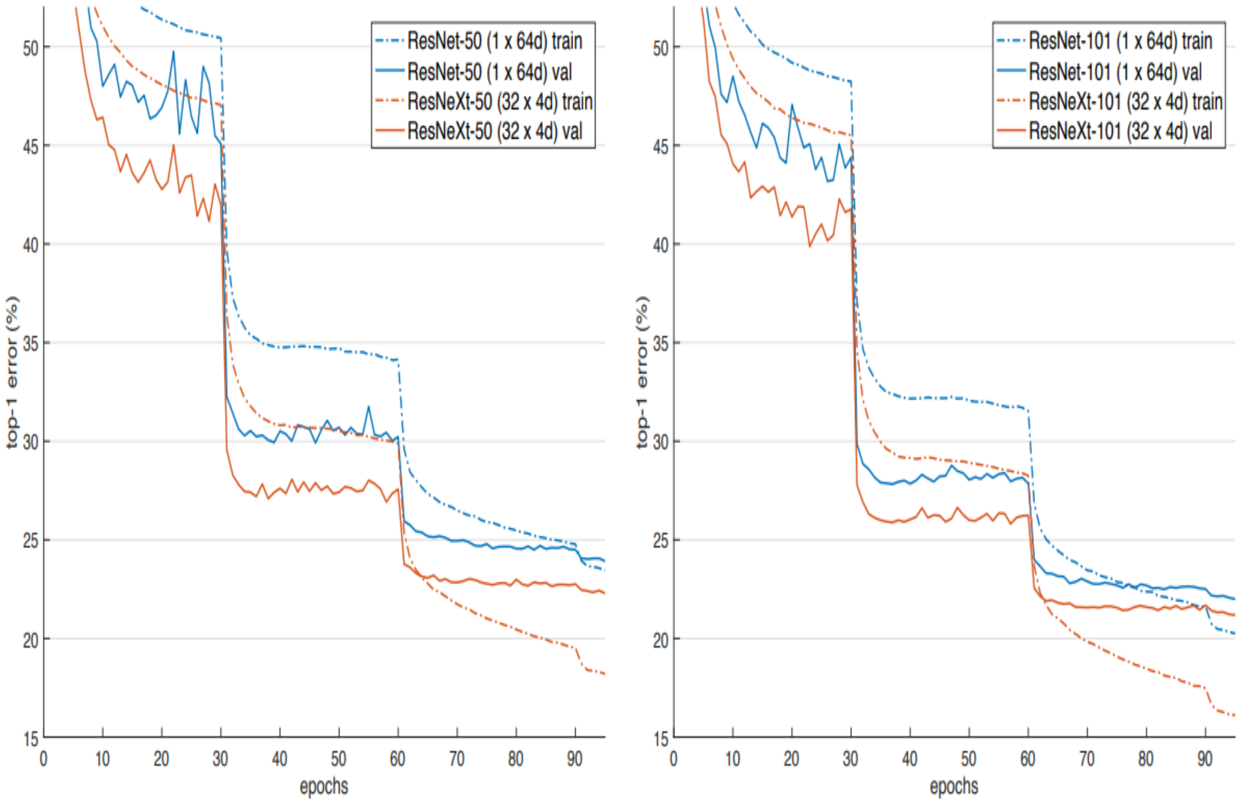
cardinality C	1	2	4	8	32
width of bottleneck d	64	40	24	14	4
width of group conv.	64	80	96	112	128

Aggregated Residual Transformations for Deep Neural Networks (CVPR 2017)

- **Experiment:**

  Cardinality라는 새로운 차원을 활용해, 기존의 ResNet 보다 더 좋은 성능을 달성

Training curves on ImageNet-1K



Ablation experiments on ImageNet-1K

	setting	top-1 error (%)
ResNet-50	1 × 64d	23.9
ResNeXt-50	2 × 40d	23.0
ResNeXt-50	4 × 24d	22.6
ResNeXt-50	8 × 14d	22.3
ResNeXt-50	32 × 4d	**22.2**
ResNet-101	1 × 64d	22.0
ResNeXt-101	2 × 40d	21.7
ResNeXt-101	4 × 24d	21.4
ResNeXt-101	8 × 14d	21.3
ResNeXt-101	32 × 4d	**21.2**

1. ResNet 보다 ResNeXt가 더 좋은 성능을 보인다.
2. Cardinality(C)와 width(d)의 설정값에 따른 비교, 32 x 4d가 가장 좋은 성능을 보인다.

Aggregated Residual Transformations for Deep Neural Networks (CVPR 2017)

- **Experiment:**
  Cardinality라는 새로운 차원을 활용해, 기존의 ResNet 보다 더 좋은 성능을 달성

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
ResNet-101 [14]	22.0	6.0	-	-
ResNet-200 [15]	21.7	5.8	20.1	4.8
Inception-v3 [39]	-	-	21.2	5.6
Inception-v4 [37]	-	-	20.0	5.0
Inception-ResNet-v2 [37]	-	-	19.9	4.9
ResNeXt-101 (**64** × **4d**)	20.4	5.3	**19.1**	**4.4**

개인적으로, 본 논문은 depth와 width를 함께 활용한 방법이란 생각이 든다.
-> cardinality는 결국 convolution을 얼마나 쪼갤지를 나타내는 것 -> Residual block을 wider하게 만든다.