Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, Kaiming He
UC San Diego; Facebook AI Research

# Aggregated Residual Transformations for Deep Neural Networks

Huidong Xie

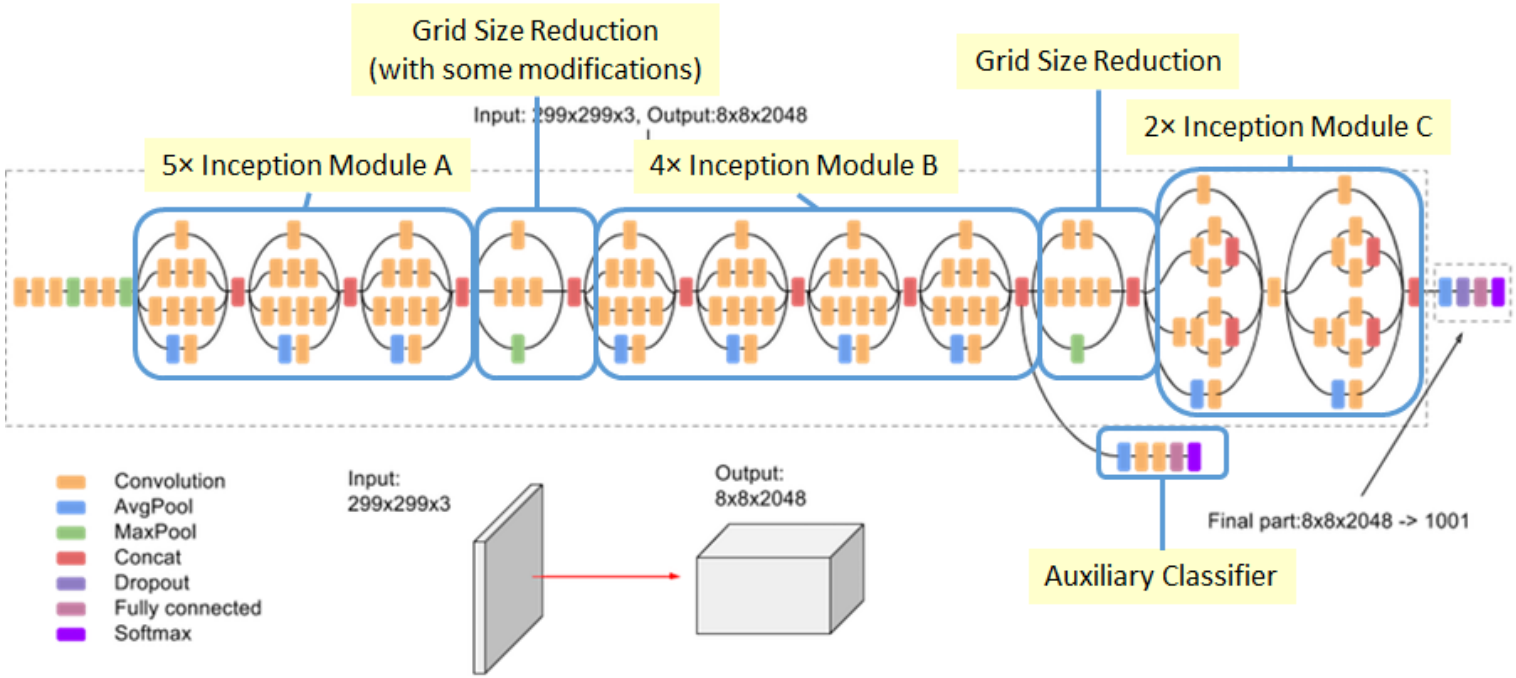
# Contributions

- They present a simple, highly modularized, multi-branch network architecture for image classification. (adopts ResNet/VGG stratergy)
- Other than width and depth, they introduce a new dimension in neural network, called cardinality, referred to as the size of the set of transformations.
- Increasing cardinality is more effective than going deeper or wider when they increase the network capacity.
- Their network, named ResNeXt, outperforms ResNet-101/152/200, Inception-v3, and inception-ResNet-v2

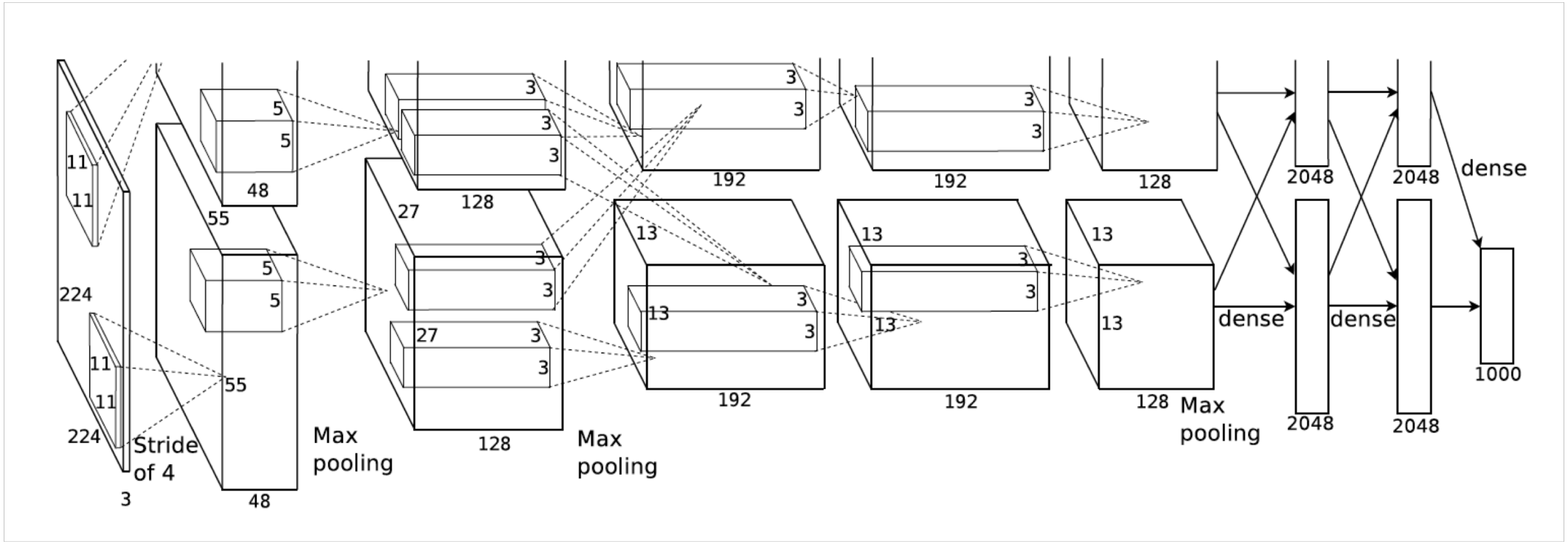
# Related work: Different network for image classifications (ResNet)

| stage | output | ResNet-50 |
|---|---|---|
| conv1 | 112×112 | 7×7, 64, stride 2 |
| conv2 | 56×56 | 3×3 max pool, stride 2<br>$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$ |
| conv3 | 28×28 | $\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$ |
| conv4 | 14×14 | $\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 6$ |
| conv5 | 7×7 | $\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$ |
| | 1×1 | global average pool<br>1000-d fc, softmax |

# Related work: Different network for image classifications (Inception network)



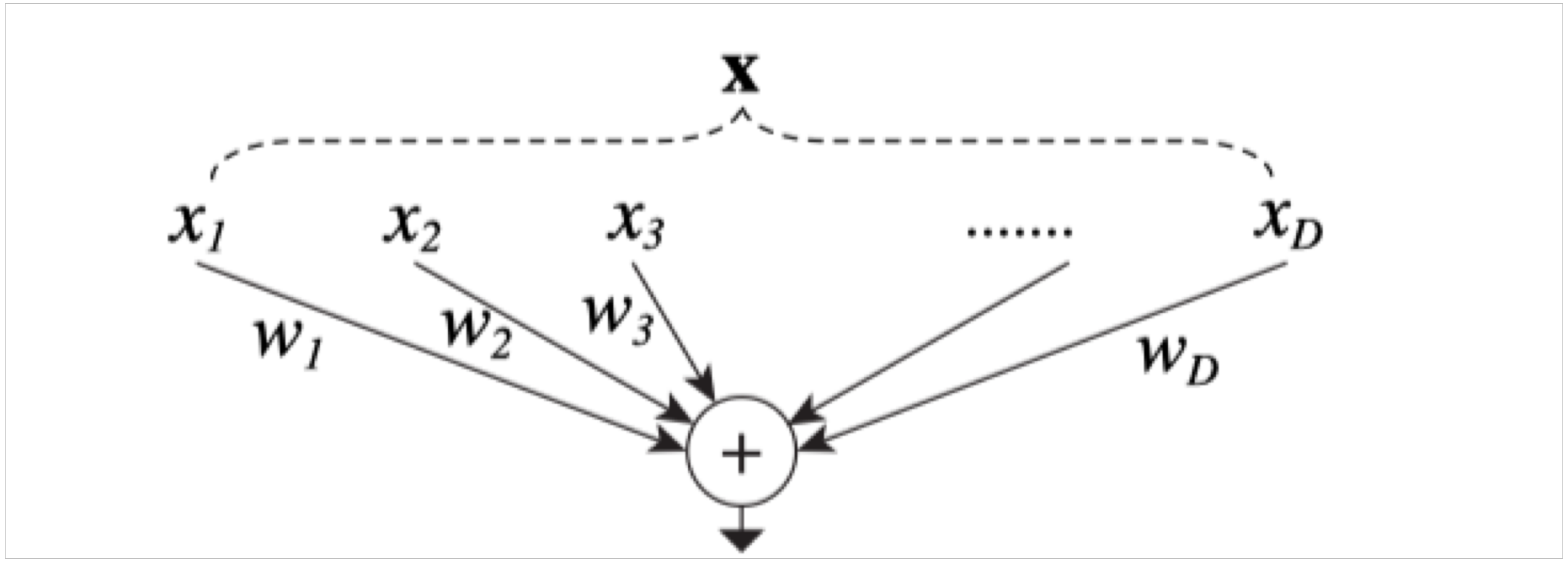
# Related work: Different network for image classifications (Group convolution)



# Compressing convolutional networks/ Ensembling

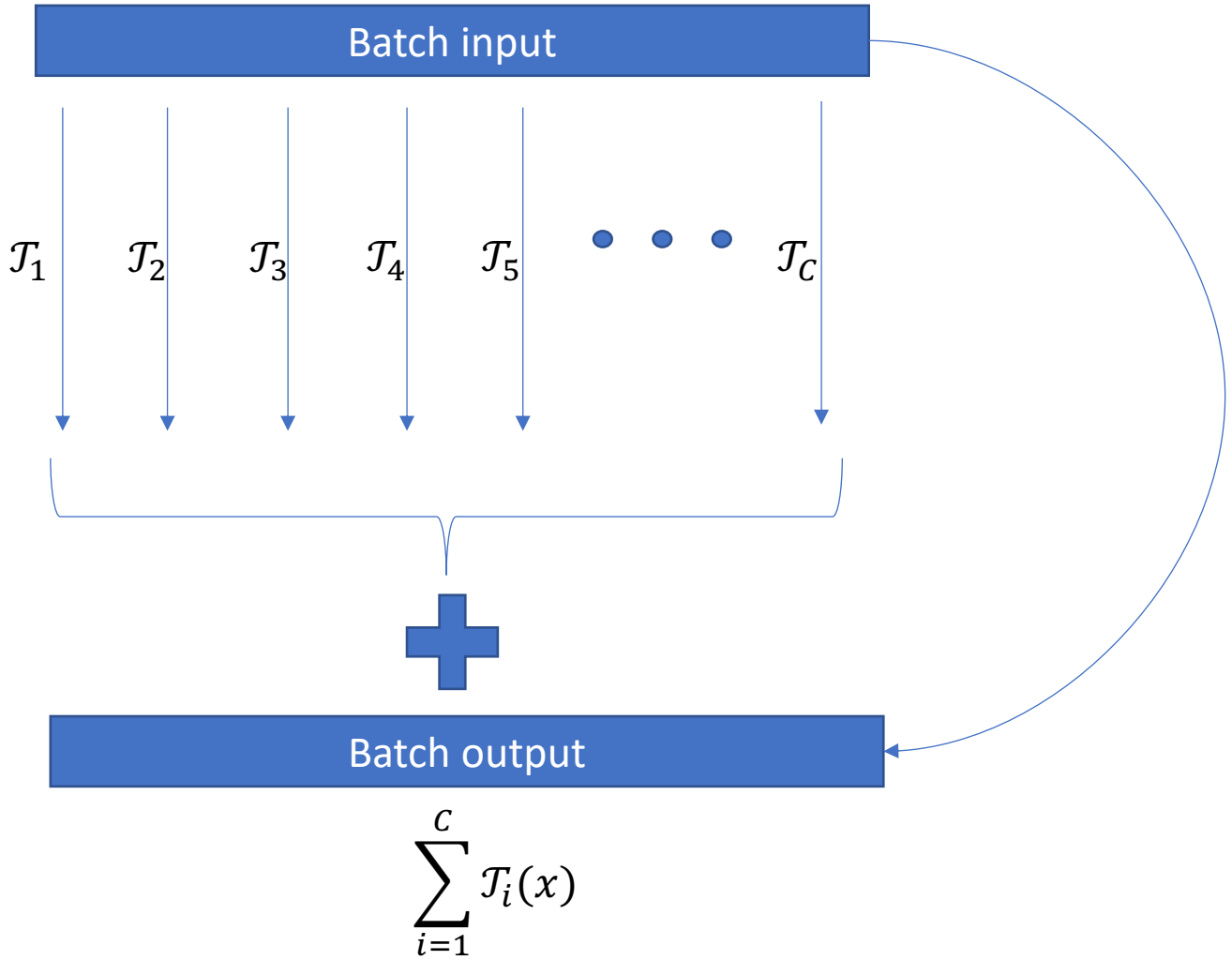
- Decomposition is a widely adopted technique to reduce the redundancy of deep convolutional network. Compressing convolutional networks show elegant compromise of accuracy with lower complexity. They claim that their network shows stronger representational power.
- Ensembling: averaging a set of independently trained networks could effectively accuracy. They argue that their network is not simply ensembling, because the members to be aggregated are trained jointly, but not independently.

# Simple neuron for inner product



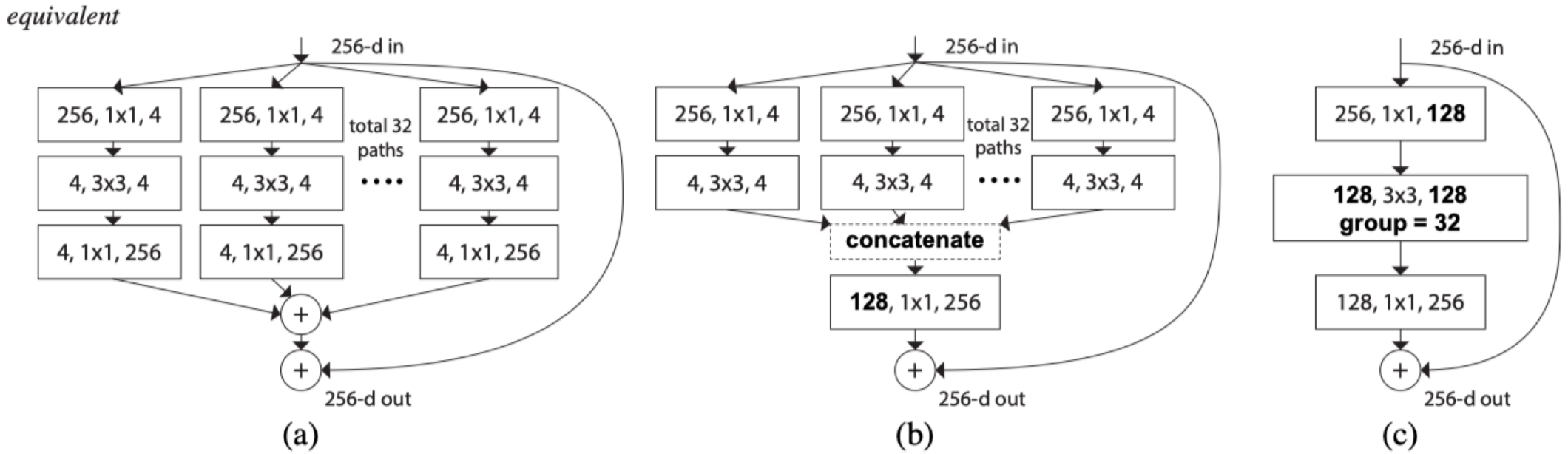

$$\sum_{i=1}^{D} w_i x_i$$

# Aggregated transformations



# $\mathcal{T}_i(x)$ can be any functions and it can serves as the residual function

ResNeXt | Inception net | Group conv

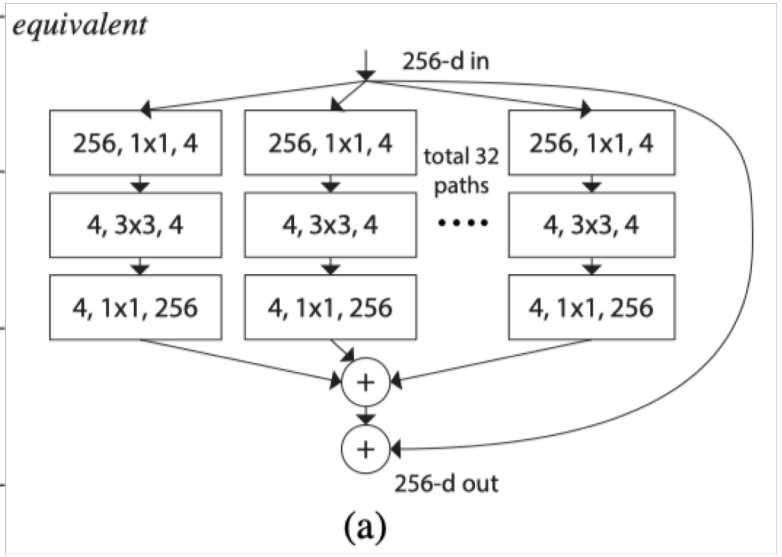



$$y = x + \sum_{i=1}^{C} \mathcal{T}_i(x)$$

# Typical ResNeXt

1. If producing spatial maps of the same size, the blocks share the same hyper-parameters
2. Each time when the spatial map is downsampled by a factor of 2, the width of the blocks is multiplied by a factor of 2.

| 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 |
| | | $\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times3$ | $\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128, C{=}32 \\ 1\times1, 256 \end{bmatrix} \times3$ |
| conv3 | 28×28 | $\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times4$ | $\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256, C{=}32 \\ 1\times1, 512 \end{bmatrix} \times4$ |
| conv4 | 14×14 | $\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times6$ | $\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512, C{=}32 \\ 1\times1, 1024 \end{bmatrix} \times6$ |
| conv5 | 7×7 | $\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times3$ | $\begin{bmatrix} 1\times1, 1024 \\ 3\times3, 1024, C{=}32 \\ 1\times1, 2048 \end{bmatrix} \times3$ |
| | 1×1 | global average pool<br>1000-d fc, softmax | global average pool<br>1000-d fc, softmax |
| # params. | | $\mathbf{25.5}\times10^6$ | $\mathbf{25.0}\times10^6$ |
| FLOPs | | $\mathbf{4.1}\times10^9$ | $\mathbf{4.2}\times10^9$ |



# Calculating # of parameters

| 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 |
| | | [ 1×1, 64<br>3×3, 64<br>1×1, 256 ] ×3 | [ 1×1, 128<br>3×3, 128, $C$=32<br>1×1, 256 ] ×3 |
| conv3 | 28×28 | [ 1×1, 128<br>3×3, 128<br>1×1, 512 ] ×4 | [ 1×1, 256<br>3×3, 256, $C$=32<br>1×1, 512 ] ×4 |
| conv4 | 14×14 | [ 1×1, 256<br>3×3, 256<br>1×1, 1024 ] ×6 | [ 1×1, 512<br>3×3, 512, $C$=32<br>1×1, 1024 ] ×6 |
| conv5 | 7×7 | [ 1×1, 512<br>3×3, 512<br>1×1, 2048 ] ×3 | [ 1×1, 1024<br>3×3, 1024, $C$=32<br>1×1, 2048 ] ×3 |
| | 1×1 | global average pool<br>1000-d fc, softmax | global average pool<br>1000-d fc, softmax |
| # params. | | **25.5**×$10^6$ | **25.0**×$10^6$ |
| FLOPs | | **4.1**×$10^9$ | **4.2**×$10^9$ |

$ResNet - 50 = 64 * 1 * 1 * 64 + 3 * 3 * 64 * 64 + 1 * 1 * 256 * 64$=57344

ResNeXt-50=(64*1*1*4+3*3*4*4+1*1*256*4)*32=45568

# Implementation details

- Input image is 224*224
- Images are down sampled by convolutional layer with stride 2
- Trained on 8 GPUs with mini batch size 256
- Trained with SGD with momentum 0.9

# Experiments on ImageNet-1K classification

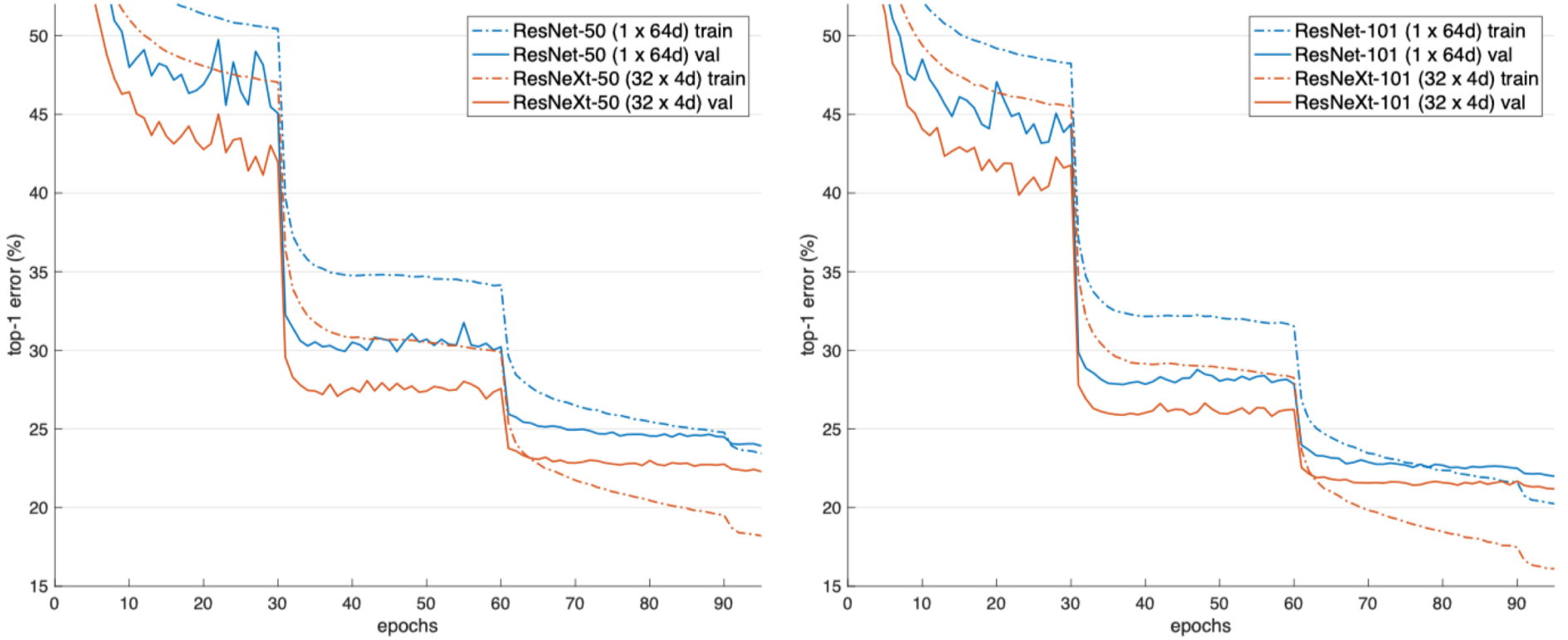



Figure 5. Training curves on ImageNet-1K. (**Left**): ResNet/ResNeXt-50 with preserved complexity (∼4.1 billion FLOPs, ∼25 million parameters); (**Right**): ResNet/ResNeXt-101 with preserved complexity (∼7.8 billion FLOPs, ∼44 million parameters).

# Increasing cardinality

| | 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** |

Table 3. Ablation experiments on ImageNet-1K. (**Top**): ResNet-50 with preserved complexity (∼4.1 billion FLOPs); (**Bottom**): ResNet-101 with preserved complexity (∼7.8 billion FLOPs). The error rate is evaluated on the single crop of 224×224 pixels.

# Increasing cardinality vs deeper/wider

ResNeXt-101 is better than deeper ResNet-200 and wider ResNet-101 with only ~50% complexity

| | 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** |

Table 4. Comparisons on ImageNet-1K when the number of FLOPs is increased to 2× of ResNet-101's. The error rate is evaluated on the single crop of 224×224 pixels. The highlighted factors are the factors that increase complexity.

# With/without residual path

| | setting | w/ residual | w/o residual |
|---|---|---|---|
| ResNet-50 | $1 \times 64$d | 23.9 | 31.2 |
| ResNeXt-50 | $32 \times 4$d | **22.2** | 26.1 |

The residual connections are helpful for optimization, where aggregated transformations are stronger representations.

# Comparisons with state-of-the-art results

| | 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** |

Table 5. State-of-the-art models on the ImageNet-1K validation set (single-crop testing). The test size of ResNet/ResNeXt is 224×224 and 320×320 as in [15] and of the Inception models is 299×299.

# Experiments on ImageNet-5k

Many models (including theirs) start to get saturated on this ImageNet-1k dataset after using multi-scale testing. They claim it is because of the complexity of the dataset but not the capability. They evaluate their models on a larger ImageNet subset.

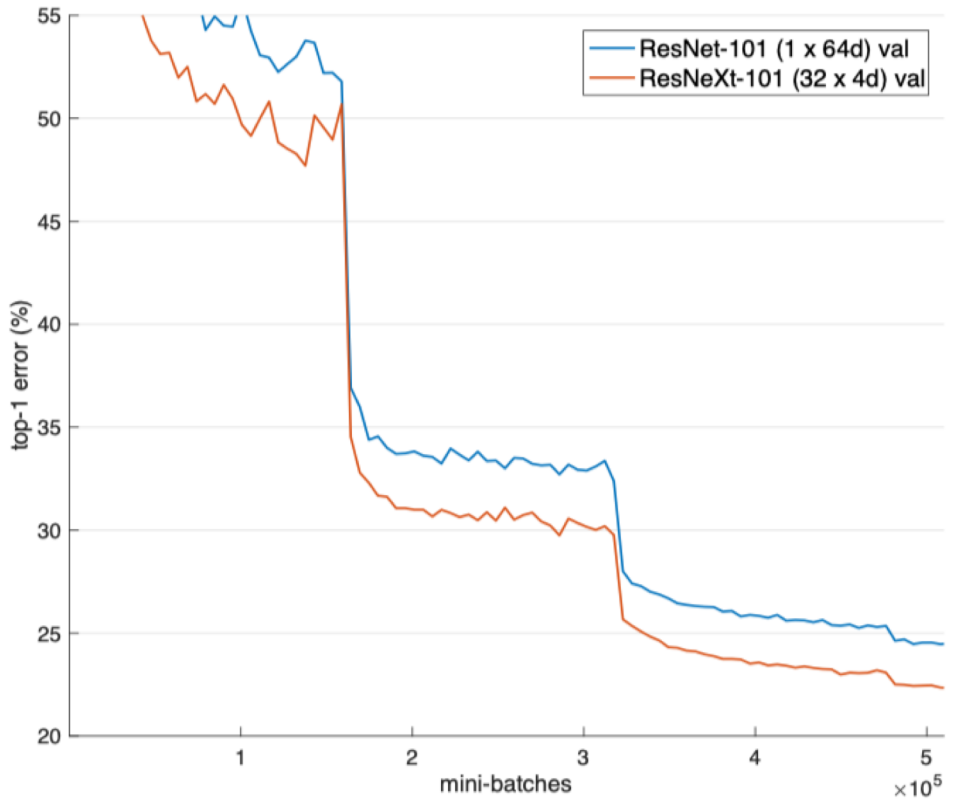



Figure 6. **ImageNet-5K** experiments. Models are trained on the 5K set and evaluated on the original 1K validation set, plotted as a 1K-way classification task. ResNeXt and its ResNet counterpart have similar complexity.

# Experiments on CIFAR-10 and 100

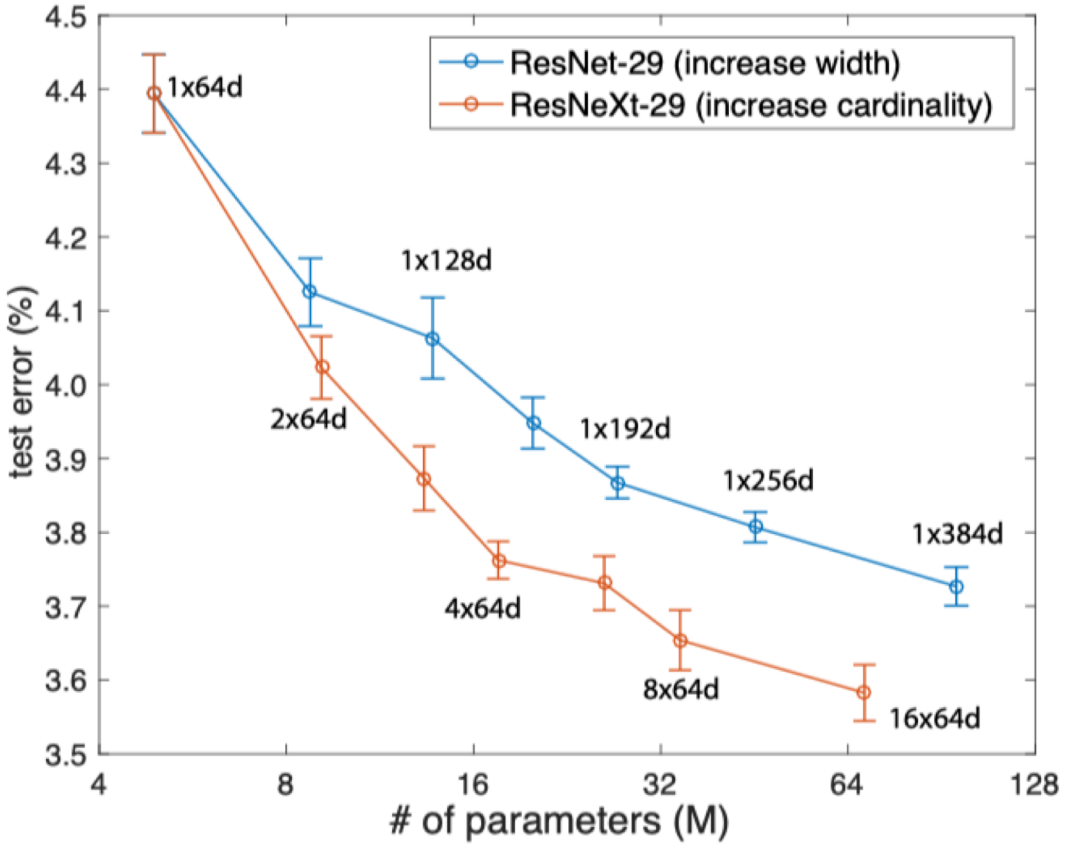



Figure 7. Test error *vs.* model size on CIFAR-10. The results are computed with 10 runs, shown with standard error bars. The labels show the settings of the templates.

# Experiments on COCO object detection

AP: average precision

| | setting | AP@0.5 | AP |
|---|---|---|---|
| ResNet-50 | 1 × 64d | 47.6 | 26.5 |
| ResNeXt-50 | 32 × 4d | **49.7** | **27.5** |
| ResNet-101 | 1 × 64d | 51.1 | 29.8 |
| ResNeXt-101 | 32 × 4d | **51.9** | **30.0** |

Table 8. Object detection results on the COCO `minival` set. ResNeXt and its ResNet counterpart have similar complexity.