

Wide Residual Networks: supplementary material

Sergey Zagoruyko
sergey.zagoruyko@enpc.fr
Nikos Komodakis
nikos.komodakis@enpc.fr

Universite Paris-Est, Ecole des Ponts
ParisTech
Paris, France

Contents

1	Network architecture	1
2	Type of convolutions in residual block	2
3	Number of convolution layers per block	2
4	Wide vs. thin residual networks	3

1 Network architecture

For clearance we provide additional schematic representation of wide residual networks in figure 1 and table 2. More detailed schematics of ResNet-10-2 and ResNet-16-2 are provided in figure 2.

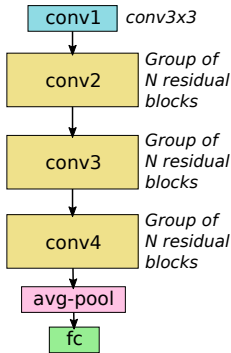


Figure 1: Schematic representation of a wide ResNet

group name	output size	block type = $B(3, 3)$
conv1	32×32	$[3 \times 3, 16]$
conv2	32×32	$\begin{bmatrix} 3 \times 3, 16 \times k \\ 3 \times 3, 16 \times k \end{bmatrix} \times N$
conv3	16×16	$\begin{bmatrix} 3 \times 3, 32 \times k \\ 3 \times 3, 32 \times k \end{bmatrix} \times N$
conv4	8×8	$\begin{bmatrix} 3 \times 3, 64 \times k \\ 3 \times 3, 64 \times k \end{bmatrix} \times N$
avg-pool	1×1	$[8 \times 8]$

Table 1: Structure of wide residual networks. Network width is determined by factor k . Original architecture [1] is equivalent to $k = 1$. Groups of convolutions are shown in brackets where N is a number of blocks in group, downsampling performed by the first layers in groups conv3 and conv4. Final classification layer is omitted for clearance. In the particular example shown, the network uses a ResNet block type $B(3, 3)$.

2 Type of convolutions in residual block

We updated table 2 (that contains residual networks with different block types) with 5-time run statistics (median, mean, std) and additional column with timings per training epoch. Block $B(3, 3)$ turned out to be the best by a little margin, and $B(3, 1)$ with $B(3, 1, 3)$ are very close to $B(3, 3)$ in accuracy having less parameters and less layers. $B(3, 1, 3)$ is faster than others by a small margin.

block type	depth	# params	time per epoch (s)	CIFAR-10
$B(1, 3, 1)$	40	1.4M	85.8	6.06 (6.06 \pm 0.16)
$B(3, 1)$	40	1.2M	67.5	5.78 (5.78 \pm 0.09)
$B(1, 3)$	40	1.3M	72.2	6.42 (6.39 \pm 0.22)
$B(3, 1, 1)$	40	1.3M	82.2	5.86 (5.87 \pm 0.09)
$B(3, 3)$	28	1.5M	67.5	5.73 (5.68 \pm 0.10)
$B(3, 1, 3)$	22	1.1M	59.9	5.78 (5.83 \pm 0.17)

Table 2: Test error (%) on CIFAR-10 of residual networks with $k = 2$ and different block types. Reported in median (mean \pm std) over 5 runs

3 Number of convolution layers per block

We updated table 3 (with varying number of convolutions per block l) with 5-time run median, mean and std statistics. As expected, the results are consistent with 1-time runs, as the differences between numbers were larger than standard deviation.

l	CIFAR-10
1	6.69 (6.75 \pm 0.08)
2	5.43 (5.47 \pm 0.12)
3	5.65 (5.64 \pm 0.19)
4	5.93 (5.95 \pm 0.22)

 Table 3: Test error (%) on CIFAR-10 of ResNet-40-2 (2.2M) with various l .

	depth- k	# params	time (ms)	CIFAR-10	CIFAR-100
VGG [†] [10]	13	20.6M	31	6.31	26.86
original-ResNet[10]	110	1.7M	-	6.43	25.16
	1202	10.2M	-	7.93	27.82
stoc-depth[10]	110	1.7M	-	5.23	24.58
	1202	10.2M	-	4.91	-
pre-act-ResNet[10]	110	1.7M	-	6.37	-
	164	1.7M	83	5.46	24.33
	1001	10.2M	512	4.64	22.71
ours	40-4	8.7M	65	4.97	22.89
	16-8	11.0M	94	4.81	22.07
	22-8	17.2M	140	4.38	21.22
	22-10	26.8M	235	4.44	20.75
	28-10	36.5M	312	4.17	20.50

Table 4: Test error of different methods on CIFAR-10 and CIFAR-100 with moderate data augmentation. We don’t use dropout for these results. VGG network was trained by us. In second column k is a widening factor. Time is provided for forward+backward update with batch size 32 on Titan X and cudnn v5. We don’t benchmark original ResNet [10] in favor to newer [10] and [10] because it requires modifications to forward and backward update.

4 Wide vs. thin residual networks

To compare with other methods in terms of the number of parameters and accuracy we provide an additional table 4 with a column specifying number of parameters for each model. We also trained an additional ResNet-28-10 with 36.5×10^6 parameters, 3.5 times more than ResNet-1001-1, that achieves 4.17% and 20.50% on CIFAR-10 and CIFAR-100, better than our other networks, which is a new state-of-the-art on these datasets.

For reference we trained «plain» modernized VGG-style network with batch normalization and average pooling on top similar to our ResNet-16-8 architecture. This network with only 13 layers achieves competitive results on CIFAR-10 and CIFAR-100. In terms of the number of parameters it is similar to ResNet-22-8, however achieves about 2% and 5% worse accuracy.

We can see that wide residual networks are much more efficient than very deep thin networks.

References

- [1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

- [2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. *CoRR*, abs/1603.05027, 2016.
- [3] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q. Weinberger. Deep networks with stochastic depth. *CoRR*, abs/1603.09382, 2016.
- [4] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

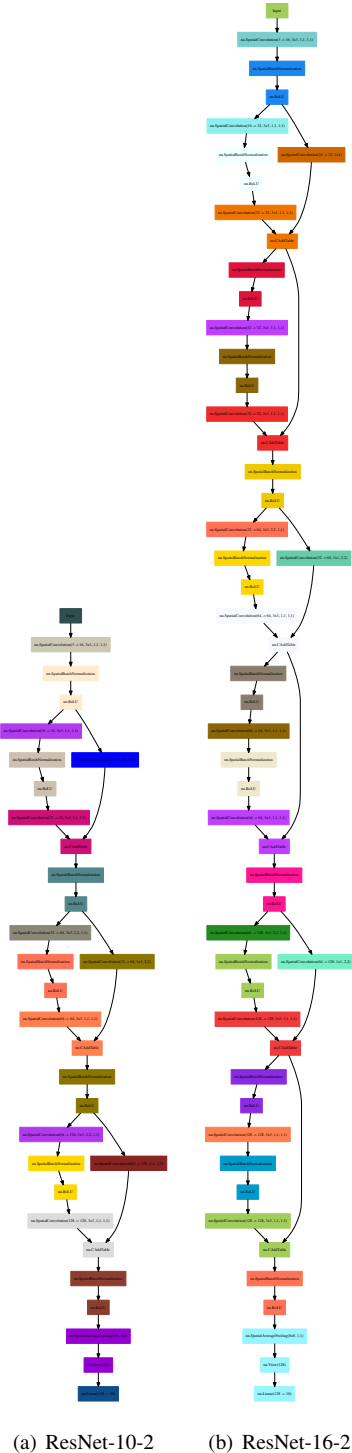


Figure 2: Schematic representations of wide residual networks. Colors mean tensor sharing between modules.