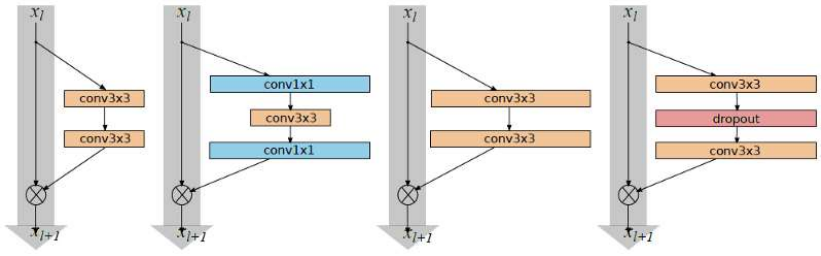
Review: WRNs — Wide Residual Networks (Image Classification)

SH Tsang Follow
Dec 1, 2018 · 5 min read

This time, **WRNs (Wide Residual Networks)** is presented. By **widening Residual Network (ResNet)**, the network can be shallower with the same accuracy or improved accuracy. **Shallower network** means:

- **Number of layers can be reduced.**
- **Training time can be shorter as well.**

A better dropout is also investigated. This is a **2016 BMVC** paper with more than **700 citations**. Though this is a paper in the year of 2016, they still keep updating the paper in June 2017. (SH Tsang @ Medium)



Various ResNet Blocks

. . .

What Are Covered

1. **Problems on Residual Network (ResNet)**
2. **WRNs (Wide Residual Networks)**
3. **Results**

. . .

1. Problems on Residual Network (ResNet)

1.1. Circuit Complexity Theory

The **circuit complexity theory** literature showing that:

> *shallow circuits can require exponentially more components than deeper circuits.*

The authors of residual networks tried to make them **as thin as possible in favor of increasing their depth and having less parameters**, and even introduced a «bottleneck» block which makes ResNet blocks even thinner.

1.2. Diminishing Feature Reuse

However, As gradient flows through the network there is nothing to force it to go through residual block weights and it can avoid learning anything during training, so it is possible that there is either **only a few blocks that learn useful representations**, or **many blocks share very little information with small contribution** to the final goal. This problem was formulated as **diminishing feature reuse**.

. . .

2. WRNs (Wide Residual Networks)

In WRNs, plenty of parameters are tested such as the design of the ResNet block, how deep (deepening factor l) and how wide (widening factor k) within the ResNet block.

When $k=1$, it has the same width of *ResNet*. While $k>1$, it is k time wider than *ResNet*.

WRN-*d*-*k*: means the WRN has the depth of d and with widening factor k.

- *Pre-Activation ResNet* is used in CIFAR-10, CIFAR-100 and SVHN datasets. Original *ResNet* is used in ImageNet dataset.
- The major difference is that *Pre-Activation ResNet* has a structure of performing batch norm and ReLU before convolution (i.e. BN-ReLU-Conv) while original *ResNet* has a structure of Conv-BN-ReLU. And *Pre-Activation ResNet* is generally better than the original one, but it has no obvious improvement in ImageNet when layers are only around 100.

2.1. The design of ResNet block

block type	depth	# params	time,s	CIFAR-10
$B(1,3,1)$	40	1.4M	85.8	6.06
$B(3,1)$	40	1.2M	67.5	5.78
$B(1,3)$	40	1.3M	72.2	6.42
$B(3,1,1)$	40	1.3M	82.2	5.86
$B(3,3)$	28	1.5M	67.5	5.73
$B(3,1,3)$	22	1.1M	59.9	5.78

WRN-d-2 (k=2), Error Rate (%) in CIFAR-10 Dataset

- **B(3;3)**: Original «basic» block, in the first figure (a)
- **B(3;1;3)**: With one extra 1×1 layer in between two 3×3 layers
- **B(1;3;1)**: With the same dimensionality of all convolutions, «straightened» **bottleneck**
- **B(1;3)**: The network has alternating 1×1, 3×3 convolutions
- **B(3;1)**: The network has alternating 3×3, 1×1 convolutions
- **B(3;1;1)**: Network-in-Network style block

B(3;3) has the smallest error rate (5.73%).

Note: Number of depths (layers) are different is to keep the number of parameters close to each other.

2.2. Number of Convolutional Layers Within ResNet block

l	CIFAR-10
1	6.69
2	5.43
3	5.65
4	5.93

WRN-40–2 with different l, Error Rate (%) in CIFAR-10 Dataset

And two 3×3 convolutions, i.e. B(3,3) has the smallest error rate than the others . Because as all networks need to be kept as near the same parameters, **B(3,3,3) and B(3,3,3,3)** turns out to have **fewer skip connection** which makes accuracy dropped. And **B(3)** has only one 3×3 convolution which **makes the feature extraction ineffective** within such shallow network within the ResNet block.

Thus, **B(3,3) is optimal** and gonna be used in the coming experiments.

2.3. Width of ResNet Blocks

depth	k	# params	CIFAR-10	CIFAR-100
40	1	0.6M	6.85	30.89
40	2	2.2M	5.33	26.04
40	4	8.9M	4.97	22.89
40	8	35.7M	4.66	-
28	10	36.5M	**4.17**	20.50
28	12	52.5M	4.33	**20.43**
22	8	17.2M	4.38	21.22
22	10	26.8M	4.44	20.75
16	8	11.0M	4.81	22.07
16	10	17.1M	4.56	21.59

Different Width (k) and Depth on CIFAR-10 and CIFAR-100

- All networks with 40, 22 and 16 layers see consistent gains when width is increased by 1 to 12 times.
- On the other hand, when keeping the same fixed widening factor k = 8 or k = 10 and varying depth from 16 to 28 there is a consistent improvement, however when we further increase depth to 40 accuracy decreases.
- Based on the results above, three sets of WRNs are chosen for comparison with state-of-the-art approaches.

. . .

3. Results

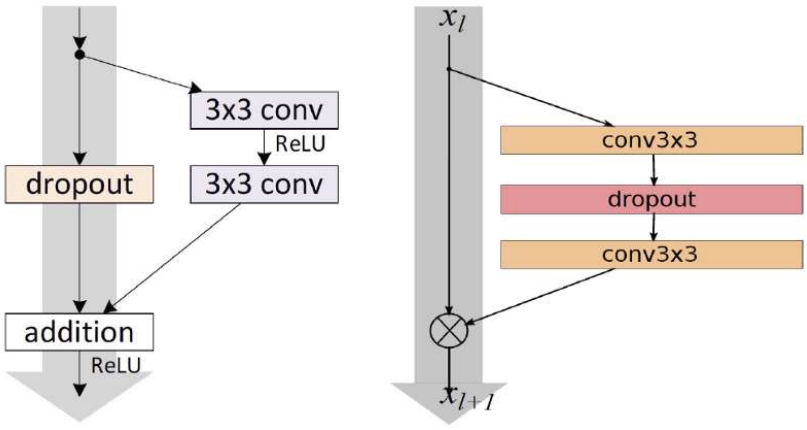
3.1. CIFAR-10 & CIFAR-100

	depth-k	# params	CIFAR-10	CIFAR-100
NIN [20]			8.81	35.67
DSN [19]			8.22	34.57
FitNet [24]			8.39	35.04
Highway [28]			7.72	32.39
ELU [5]			6.55	24.28
original-ResNet[11]	110	1.7M	6.43	25.16
	1202	10.2M	7.93	27.82
stoc-depth[14]	110	1.7M	5.23	24.58
	1202	10.2M	4.91	-
pre-act-ResNet[13]	110	1.7M	6.37	-
	164	1.7M	5.46	24.33
	1001	10.2M	4.92(4.64)	22.71
WRN (ours)	40-4	8.9M	4.53	21.18
	16-8	11.0M	4.27	20.43
	28-10	36.5M	**4.00**	**19.25**

CIFAR-10 & CIFAR-100

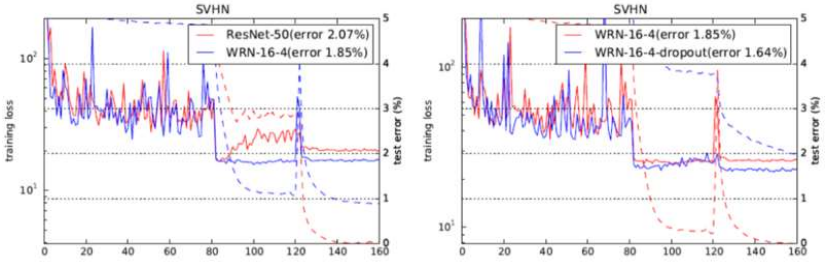
- **WRN-40–4**: Fewer parameters (8.9M) than 1001-layer *Pre-Activation ResNet* (10.2M). But it also got **lower error rate** as well. (4.52% on CIFAR-10 and 21.18% on CIFAR-100)
- **WRN-16-8 & WRN-28-10**: Shallower than and wider than WRN-40–4, and got even lower error rate. **With shallower network, training time can be shorter since parallel computations are performed on GPUs no matter how wide.**
- And it is **the first paper to obtain lower than 20% for CIFAR-100 without any strong data augmentation!!!**

3.2. Dropout



Dropout in Original ResNet (Left) and Dropout in WRNs (Right)

depth	k	dropout	CIFAR-10	CIFAR-100	SVHN
16	4		5.02	24.03	1.85
16	4	✓	5.24	23.91	1.64
28	10		4.00	19.25	-
28	10	✓	**3.89**	**18.85**	-
52	1		6.43	29.89	2.08
52	1	✓	6.28	29.78	1.70



Dropout Is Better

- Top: With dropout, consistent gain is obtained for different depth, k, and datasets.
- Bottom right: With dropout, the training loss is higher but test error is lower meaning that dropout reduce overfitting successfully.

3.3. ImageNet & COCO

width		1.0	1.5	2.0	3.0
WRN-18	top1,top5	30.4, 10.93	27.06, 9.0	25.58, 8.06	24.06, 7.33
	#parameters	11.7M	25.9M	45.6M	101.8M
WRN-34	top1,top5	26.77, 8.67	24.5, 7.58	23.39, 7.00	
	#parameters	21.8M	48.6M	86.0M	

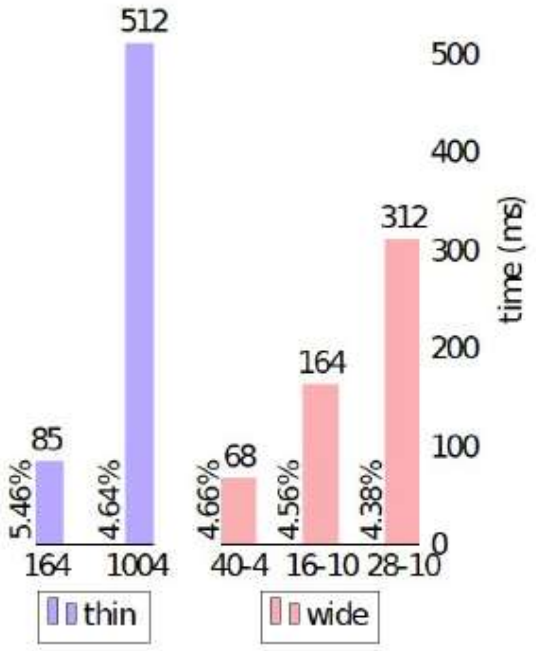
Single Crop Single Model Validation Error, ImageNet

- The above networks obtain similar accuracy than the original one with 2 times fewer layers.

Dataset	model	dropout	test perf.
CIFAR-10	WRN-40-10	✓	3.8%
CIFAR-100	WRN-40-10	✓	18.3%
SVHN	WRN-16-8	✓	1.54%
ImageNet (single crop)	WRN-50-2-bottleneck		21.9% top-1, 5.79% top-5
COCO test-std	WRN-34-2		35.2 mAP

- **WRN-50–2-Bottleneck**: Outperforms ResNet-152 and having 3 times fewer layers, which means the training time is significantly faster.
- **WRN-34–2**: Outperforms ResNet-152 and Inception-v4-based models

3.4. Training Time



Training Time for Each Batch with Batch Size of 32, CIFAR-10

- **WRN-16–10 and WRN-28–10**: The training time is much lower than the 1004-layer *Pre-Activation ResNet*, and having lower error rate.
- **WRN-40–4**: The training time is lower than the 164-layer *Pre-Activation ResNet*, and having lower error rate.

. . .

Since training takes much time, it can take couples of days or even weeks. When the training set is gonna larger and larger, a better way to train is needed. Indeed, in recent research, many researchers are still focusing on how to reduce the training time or number of epoches for training.

In WRNs, it reduce the training time but with the expense of increasing the number of parameters due to the widening of the network.

. . .

References

[2016 BMVC] [WRNs]
Wide Residual Networks

My Related Reviews on Image Classification

[LeNet] [AlexNet] [ZFNet] [VGGNet] [SPPNet] [PReLU-Net] [GoogLeNet / Inception-v1] [BN-Inception / Inception-v2] [Inception-v3] [Inception-v4] [Xception] [MobileNetV1] [ResNet] [Pre-Activation ResNet] [RiR] [Stochastic Depth] [DenseNet]