Ideas for improving deep residual learning

Jiyuu Yi

References

- K. He, X. Zhang, S. Ren and J. Sun. Identity Mapping in Deep Residual Networks. arXiv 2016.
- G. Huang and Y. Sun. Deep Networks with Stochastic Depth. arXiv 2016.
- K. Zhang, M. Sun and T.X. Han. Residual Networks of Residual Networks: Multilevel Residual Networks. arXiv 2016

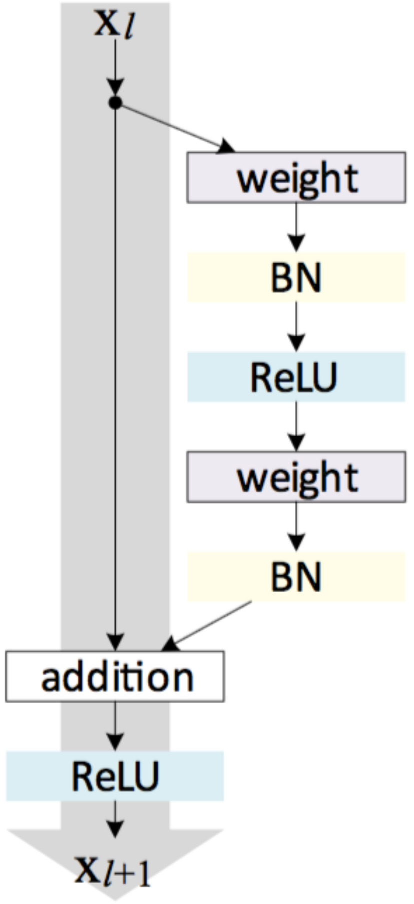
Identity Mapping with Deep Residual Networks

K. He, X. Zhang, S. Ren and J. Sun

arXiv 2016

Deep Residual Networks

- Central idea of ResNets.
  - To learn the additive function $\mathcal{F}(\mathbf{x}_l, \mathcal{W}_l)$.
- Analyzing deep residual networks
  - by focusing on creating 'direct' path.
- If both $h(\mathbf{x}_l)$ and $f(\mathbf{y}_l)$ are identity mapping,
  - the signal could be directly propagated from one unit to any other units.



$$\mathbf{y}_l = h(\mathbf{x}_l) + \mathcal{F}(\mathbf{x}_l, \mathcal{W}_l),$$
$$\mathbf{x}_{l+1} = f(\mathbf{y}_l),$$

Analysis of Deep Residual Networks

$$\mathbf{y}_l = h(\mathbf{x}_l) + \mathcal{F}(\mathbf{x}_l, \mathcal{W}_l), \tag{1}$$

$$\mathbf{x}_{l+1} = f(\mathbf{y}_l), \tag{2}$$

- On original residual unit
  - $h(\mathbf{x}_l)$ is identity mapping
  - f is ReLU

- If f also identity mapping： $\mathbf{x}_{l+1} \equiv \mathbf{y}_l$
  - we can put Eqn.(1) into Eqn.(2) and obtain

$$\mathbf{x}_{l+1} = \mathbf{x}_l + \mathcal{F}(\mathbf{x}_l, \mathcal{W}_l) \tag{3}$$

Analysis of Deep Residual Networks

$$\mathbf{x}_{l+1} = \mathbf{x}_l + \mathcal{F}(\mathbf{x}_l, \mathcal{W}_l) \tag{3}$$

- Recursively we will have Eqn. (4)
  - for any deeper unit L and any shallower unit l

$$\mathbf{x}_L = \mathbf{x}_l + \sum_{i=l}^{L-1} \mathcal{F}(\mathbf{x}_i, \mathcal{W}_i), \tag{4}$$

- Eqn. (4) also lead to nice backward propagation properties

$$\frac{\partial \mathcal{E}}{\partial \mathbf{x}_l} = \frac{\partial \mathcal{E}}{\partial \mathbf{x}_L} \frac{\partial \mathbf{x}_L}{\partial \mathbf{x}_l} = \frac{\partial \mathcal{E}}{\partial \mathbf{x}_L} \left(1 + \frac{\partial}{\partial \mathbf{x}_l} \sum_{i=l}^{L-1} \mathcal{F}(\mathbf{x}_i, \mathcal{W}_i)\right). \tag{5}$$

Analysis of Deep Residual Networks

$$\frac{\partial \mathcal{E}}{\partial \mathbf{x}_l} = \frac{\partial \mathcal{E}}{\partial \mathbf{x}_L}\frac{\partial \mathbf{x}_L}{\partial \mathbf{x}_l} = \frac{\partial \mathcal{E}}{\partial \mathbf{x}_L}\left(1 + \frac{\partial}{\partial \mathbf{x}_l}\sum_{i=l}^{L-1}\mathcal{F}(\mathbf{x}_i, \mathcal{W}_i)\right). \tag{5}$$

- Eqn. (4) also lead to nice backward propagation properties.
  - A term of $\frac{\partial \mathcal{E}}{\partial \mathbf{x}_L}$ ensures that information is directly propagated back to any shollwer unit l
  - In Eqn. (5) $\frac{\partial \mathcal{E}}{\partial \mathbf{x}_l}$ is unlikely to be cancled because in the general term $\frac{\partial}{\partial \mathbf{x}_l}\sum_{i=l}^{L-1}\mathcal{F}(\mathbf{x}_i, \mathcal{W}_i)$ cannot be always -1 for all sample in mini-batch.
    - So the gradient of a layer does not vanish

Importance of Identity Skip Connections

- Let's consider a simple modification to break identity shortcut

$$h(\mathbf{x}_l) = \lambda_l \mathbf{x}_l$$

- Then we will have

$$\mathbf{x}_{l+1} = \lambda_l \mathbf{x}_l + \mathcal{F}(\mathbf{x}_l, \mathcal{W}_l), \tag{6}$$

$$\mathbf{x}_L = (\prod_{i=l}^{L-1} \lambda_i)\mathbf{x}_l + \sum_{i=l}^{L-1} (\prod_{j=i+1}^{L-1} \lambda_j)\mathcal{F}(\mathbf{x}_i, \mathcal{W}_i)$$

$$\mathbf{x}_L = (\prod_{i=l}^{L-1} \lambda_i)\mathbf{x}_l + \sum_{i=l}^{L-1} \hat{\mathcal{F}}(\mathbf{x}_i, \mathcal{W}_i), \tag{7}$$

$$\frac{\partial \mathcal{E}}{\partial \mathbf{x}_l} = \frac{\partial \mathcal{E}}{\partial \mathbf{x}_L} \left((\prod_{i=l}^{L-1} \lambda_i) + \frac{\partial}{\partial \mathbf{x}_l} \sum_{i=l}^{L-1} \hat{\mathcal{F}}(\mathbf{x}_i, \mathcal{W}_i) \right). \tag{8}$$

Importance of Identity Skip Connection

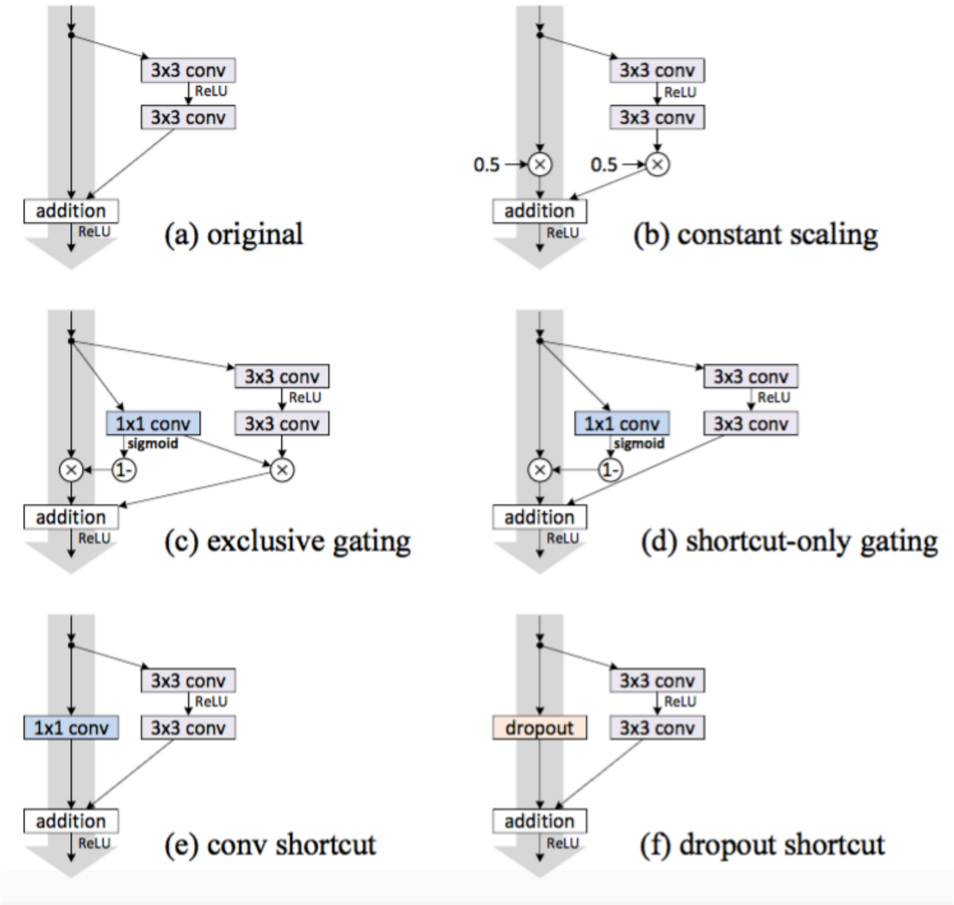
$$\frac{\partial \mathcal{E}}{\partial \mathbf{x}_l} = \frac{\partial \mathcal{E}}{\partial \mathbf{x}_L} \left(\left(\prod_{i=l}^{L-1} \lambda_i\right) + \frac{\partial}{\partial \mathbf{x}_l} \sum_{i=l}^{L-1} \hat{\mathcal{F}}(\mathbf{x}_i, \mathcal{W}_i) \right). \tag{8}$$

- By a factor $\prod_{i=l}^{L-1} \lambda_i$ in Eqn. (8)
  - if $\lambda_i > 1$ for all i, this factor can be exponentially large
  - If $\lambda_i < 1$ for all i, this factor can be exponentially small and vanish

Experiment on skip connection

Table 1. Classification error on the CIFAR-10 test set using ResNet-110 [1], with different types of shortcut connections applied to all Residual Units. We report "fail" when the test error is higher than 20%.

case	Fig.	on shortcut	on $\mathcal{F}$	error (%)	remark
original [1]	Fig. 2(a)	1	1	**6.61**	
constant scaling	Fig. 2(b)	0	1	fail	This is a plain net
		0.5	1	fail	
		0.5	0.5	12.35	frozen gating
exclusive gating	Fig. 2(c)	$1-g(\mathbf{x})$	$g(\mathbf{x})$	fail	init b_g=0 to -5
		$1-g(\mathbf{x})$	$g(\mathbf{x})$	8.70	init b_g=-6
		$1-g(\mathbf{x})$	$g(\mathbf{x})$	9.81	init b_g=-7
shortcut-only gating	Fig. 2(d)	$1-g(\mathbf{x})$	1	12.86	init b_g=0
		$1-g(\mathbf{x})$	1	6.91	init b_g=-6
1×1 conv shortcut	Fig. 2(e)	1×1 conv	1	12.22	
dropout shortcut	Fig. 2(f)	dropout 0.5	1	fail	



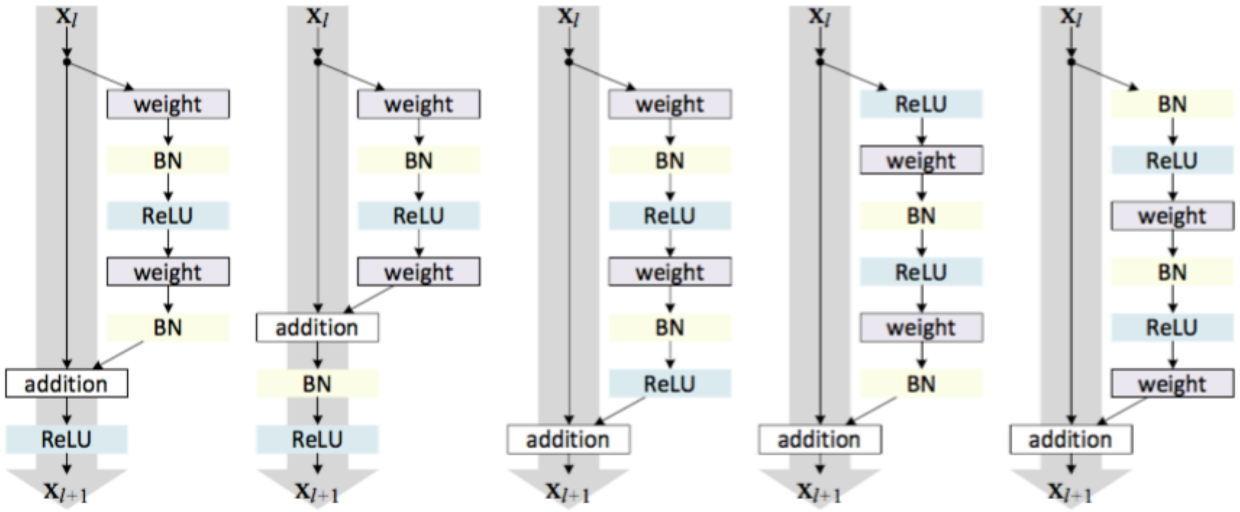
On the Usage of Activation Functions

- We assumed that f is the identity mapping in Eqn.(5) and (8).
- But in the above experiment f is ReLU.
  - So Eqn(5) and (8) are approximate in the above experiments.
- Let's investigate the impact of f, and we will make f an identity mapping

Experiment on Activation

Table 2. Classification error (%) on the CIFAR-10 test set using different activation functions.

case	Fig.	ResNet-110	ResNet-164
original Residual Unit [1]	Fig. 4(a)	6.61	5.93
BN after addition	Fig. 4(b)	8.17	6.50
ReLU before addition	Fig. 4(c)	7.84	6.14
ReLU-only pre-activation	Fig. 4(d)	6.71	5.91
full pre-activation	Fig. 4(e)	**6.37**	**5.46**



(a) original (b) BN after addition (c) ReLU before addition (d) ReLU-only pre-activation (e) **full pre-activation**

Pre-activation

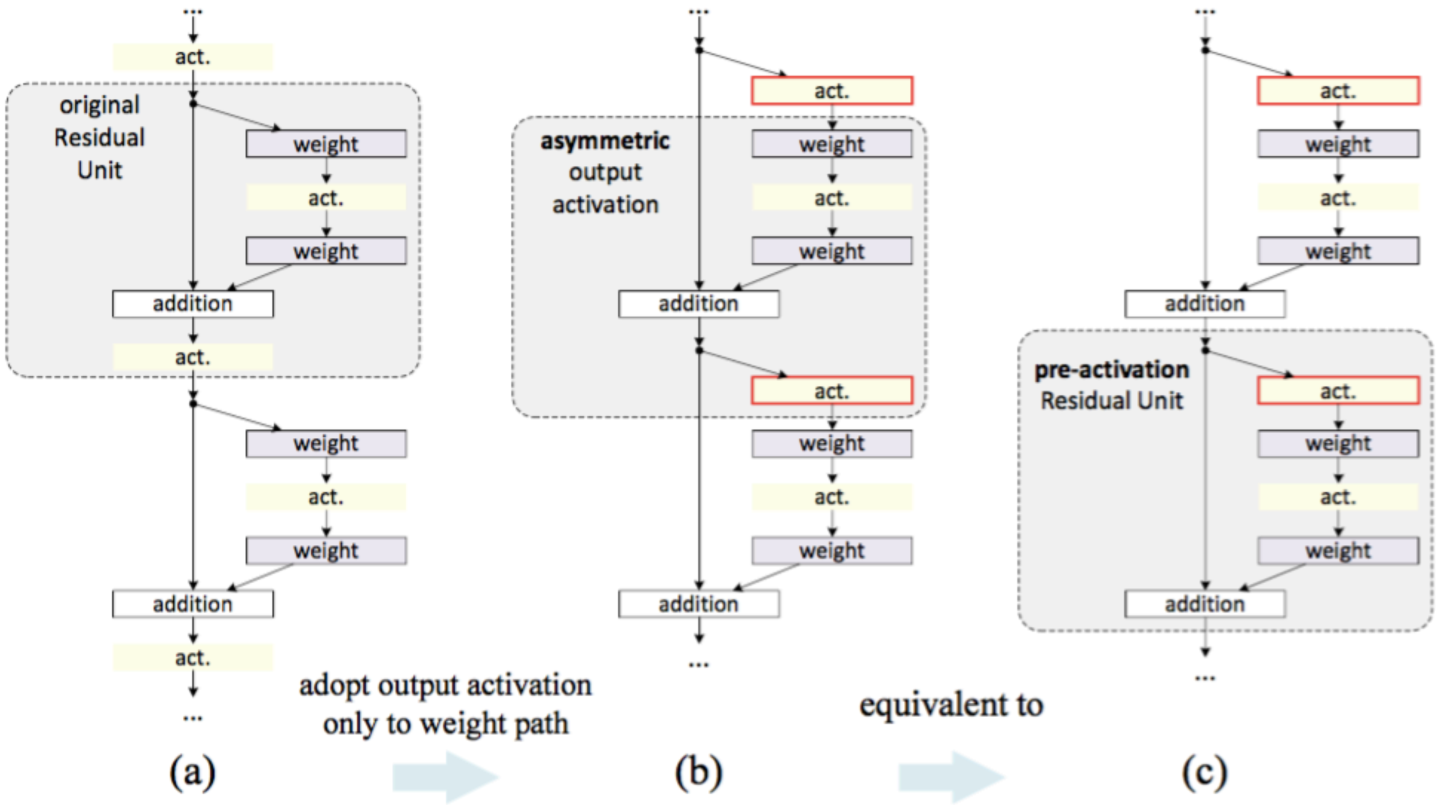
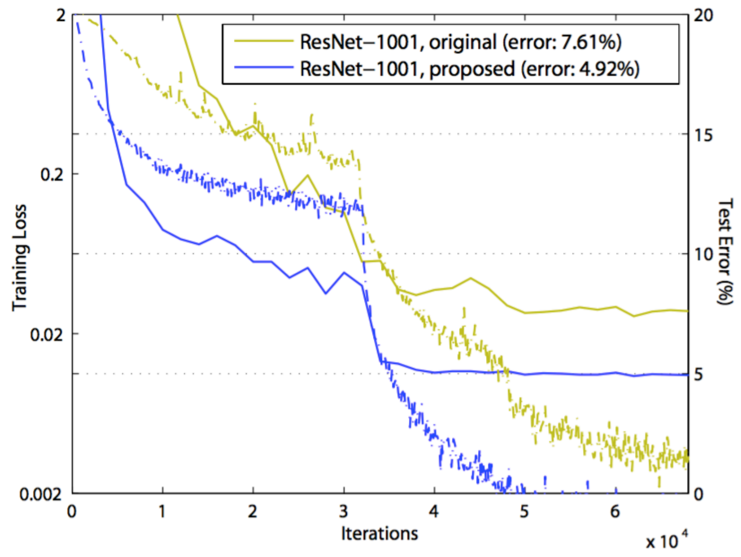
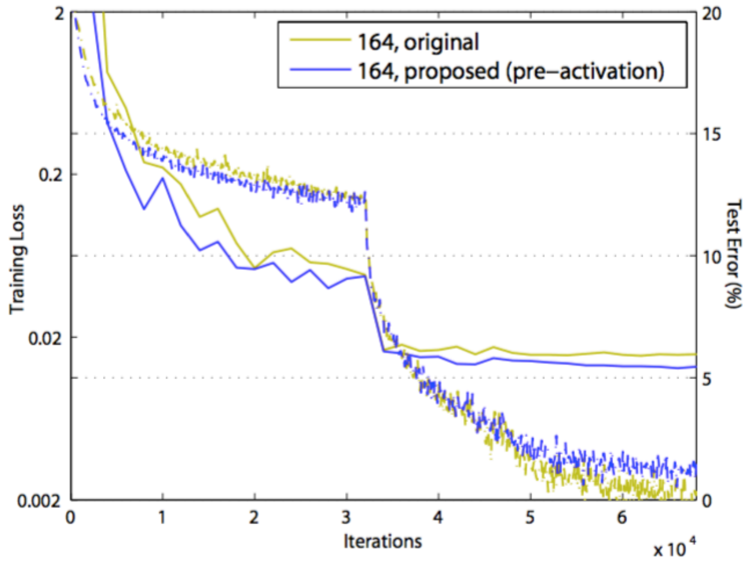


Figure 5. Using asymmetric after-addition activation is equivalent to constructing a *pre-activation* Residual Unit.

Impact of pre-activation

- Ease of optimization
  - Original design is affected by ReLU
  - But f of proposed design is identity mapping
- Reducing overfitting
  - Higher training loss
  - But lower test error
- Conclusion : Identity shortcut connection and identity after-addition activation make information propagation smooth.





Deep Networks with Stochastic Depth

G. Huang, Y. Sun and Z.Liu

arXiv 2016

Problems of Deep Networks

- Network depth is a major determenant of model expressiveness

- However very deep models also introduce new challenges
  - Vanishing gradient
  - Diminishing feature reuse
    - washed out features of input instances through repeated convolution weight matrices
  - Long training time

- Stochastic depth can alleviate these problems

Stochastic Depth

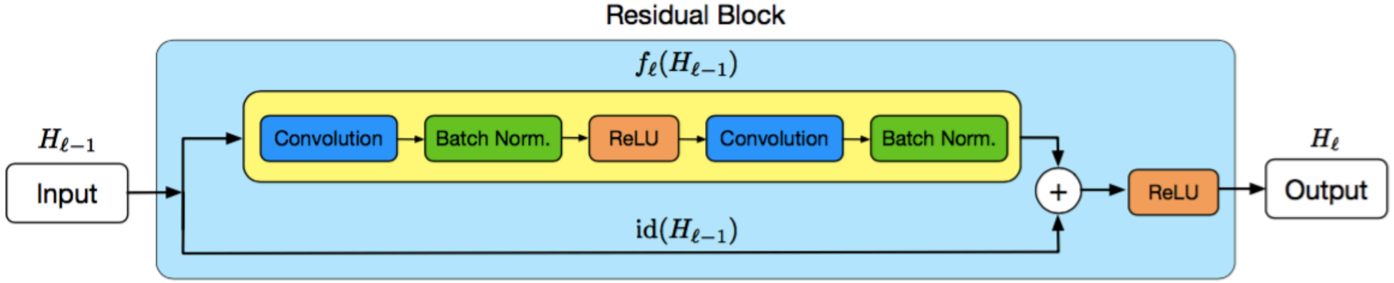
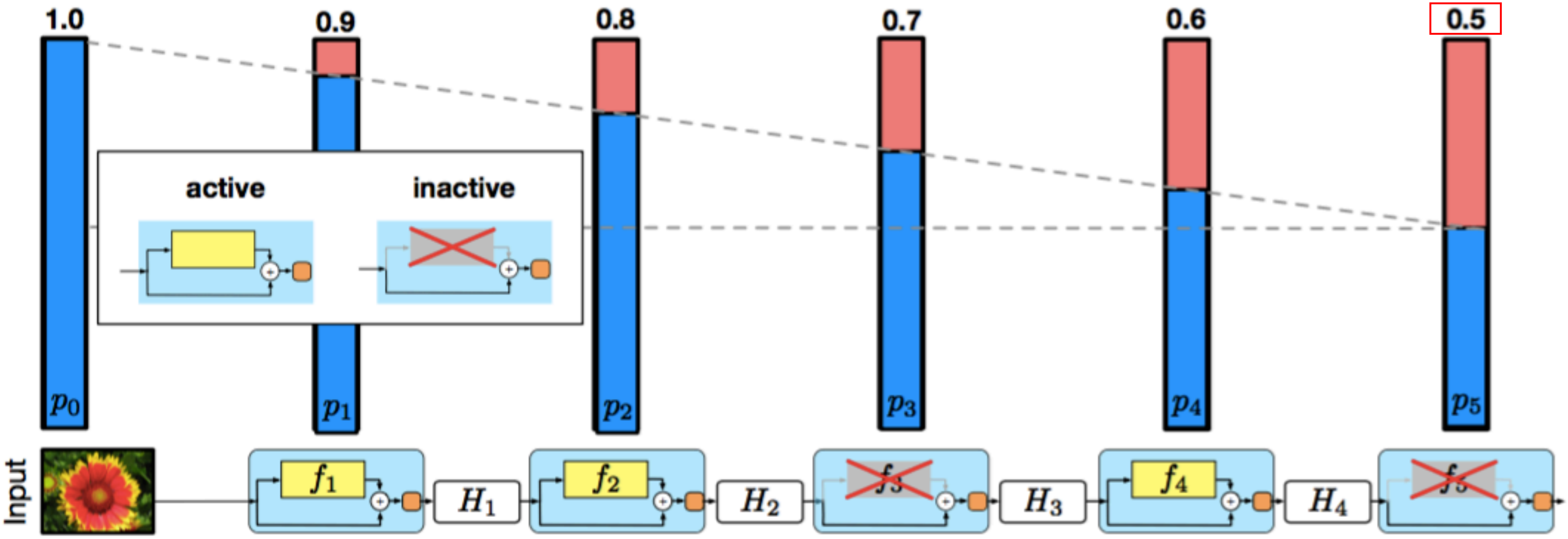


Fig. 1. A close look at the ℓ^{th} ResBlock in a ResNet.

- Original Residual Unit: $H_\ell = \mathtt{ReLU}(f_\ell(H_{\ell-1}) + \mathrm{id}(H_{\ell-1}))$
- Simple modification: $H_\ell = \mathtt{ReLU}(b_\ell f_\ell(H_{\ell-1}) + \mathrm{id}(H_{\ell-1}))$.
  - $b_\ell \in \{0, 1\}$ denotes a Bernoulli random variable.

Linearly decayed survaval probabilities

New hyper-parameter



Experiment and advantage of stochastic depth

- Reducing expected network depth: $E(\tilde{L}) = \sum_{\ell=1}^{L} p_\ell$

- Reducing training time

	CIFAR10+	CIFAR100+	SVHN
Constant Depth	20h 42m	20h 51m	33h 43m
Stochastic Depth	15h 7m	15h 20m	25h 33m

- Implicit model ensemble

- Training with an aggressively deep ResNet

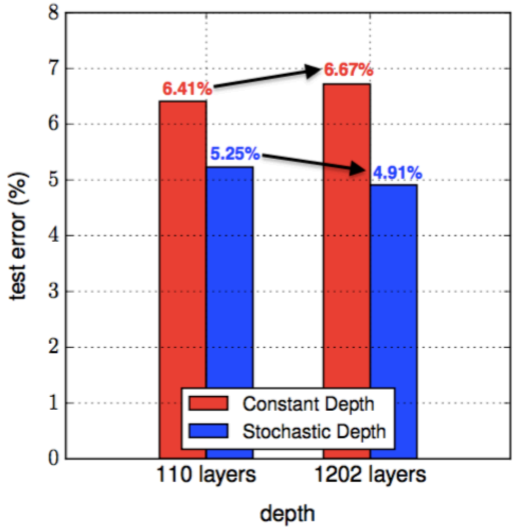


Fig. 5. With stochastic depth, the 1202-layer ResNet still significantly improves over the 110-layer one.

Experiment and Impact of stochastic depth

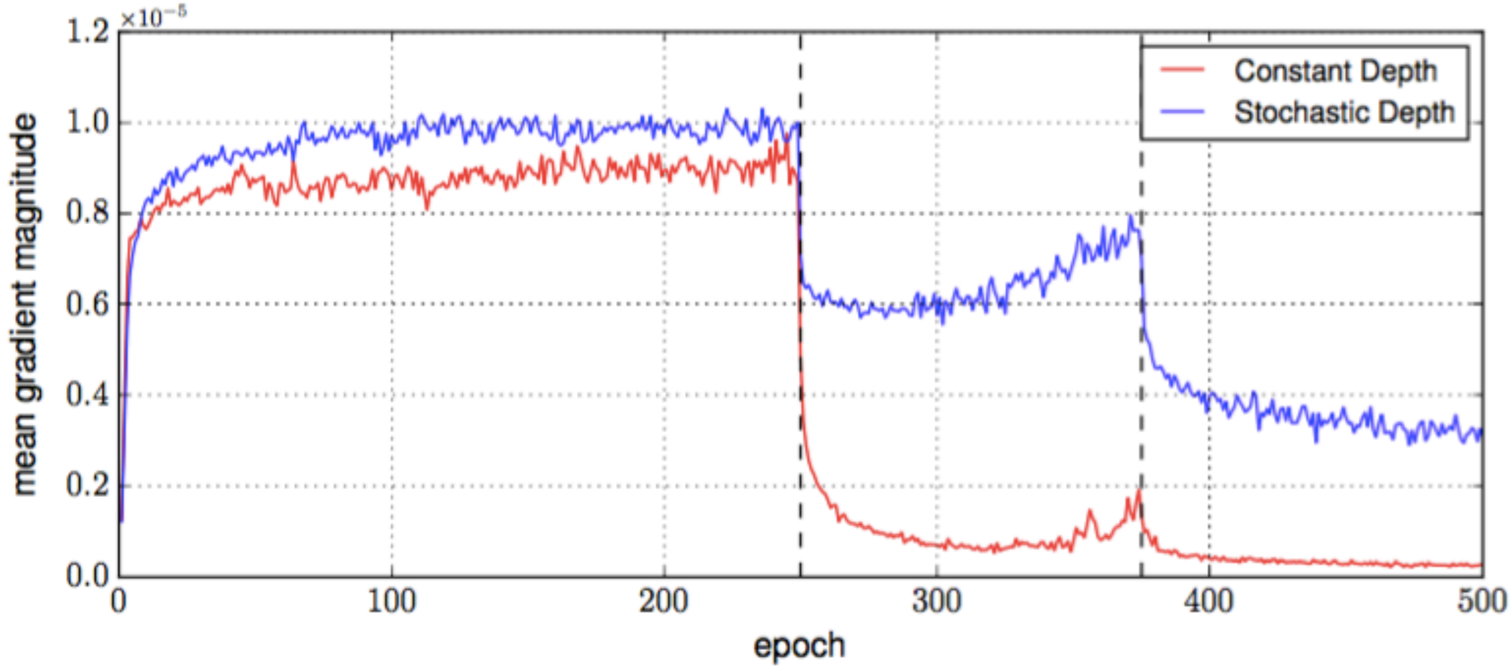


Fig. 7. The first convolutional layer's mean gradient magnitude for each epoch during training. The vertical dotted lines indicate scheduled reductions in learning rate by a factor of 10, which cause gradients to shrink.

Experiment and Impact of stochastic depth

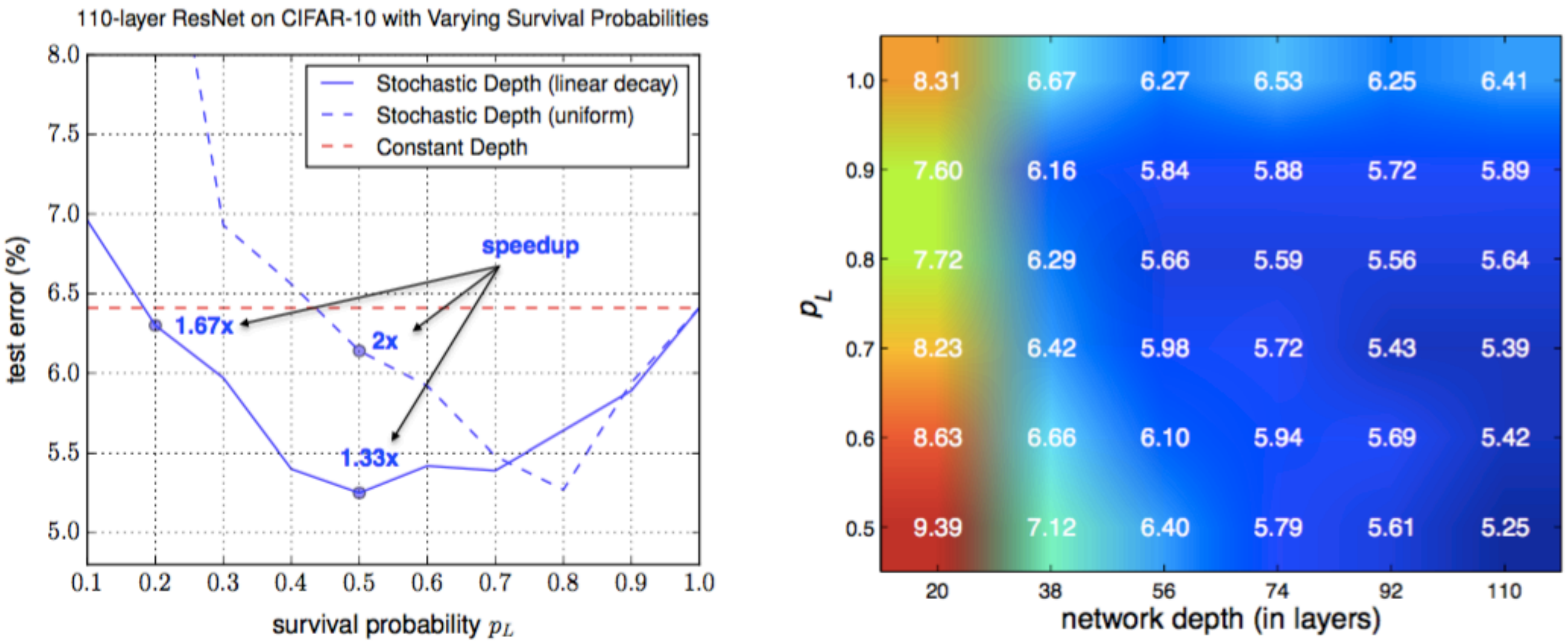


Fig. 8. Left: Test error (%) on CIFAR-10 with respect to the p_L with uniform and decaying assignments of p_ℓ. Right: Test error (%) heatmap on CIFAR-10 varyied over p_L and network depth.

Residual Networks of Residual Networks: Multilevel Residual Networks

K. Zhang, M. Sun and T.X. Han

arXiv 2016

Motivation

- Very deep models suffer from two problems
  - vanishing gradient
  - overfitting

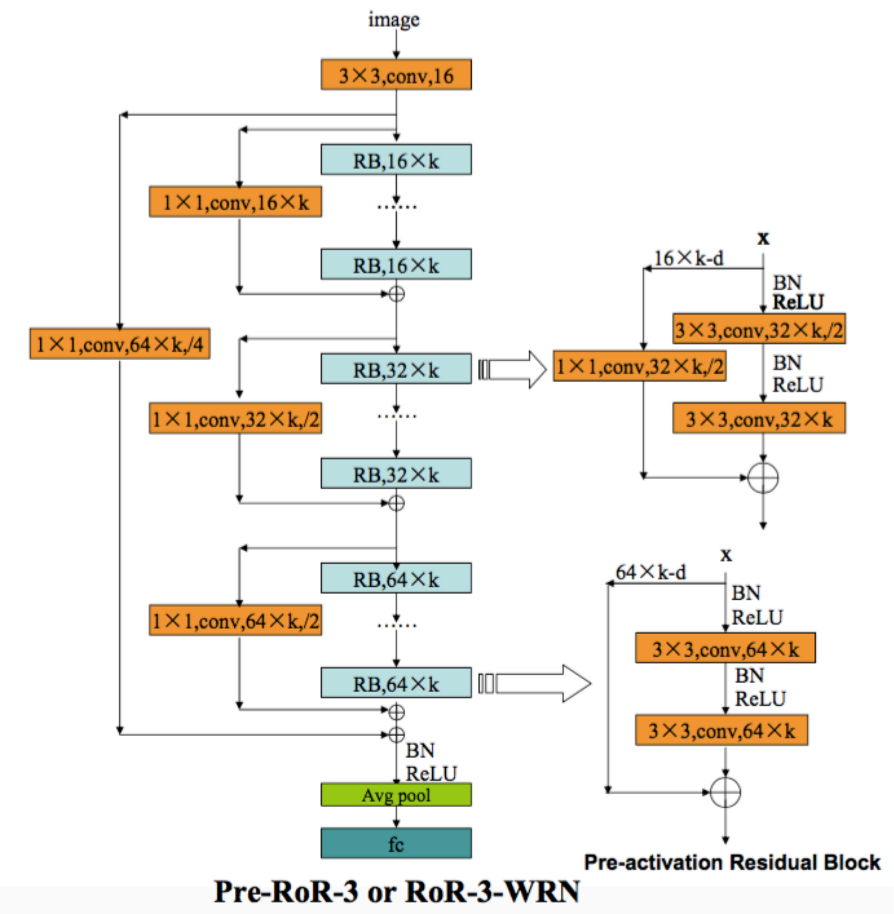
- RoR is based on a hypothesis：
  - To dig the optimization ability of residual networks, we can optimize the residual mapping of residual mapping.

Architecture of RoR-3

$$y_{L/3} = g(x_1) + h(x_{L/3}) + F(x_{L/3}, W_{L/3}),$$
$$x_{L/3+1} = f(y_{L/3})$$

$$y_{2L/3} = g(x_{L/3+1}) + h(x_{2L/3}) + F(x_{2L/3}, W_{2L/3}),$$
$$x_{2L/3+1} = f(y_{2L/3})$$

$$y_L = g(x_1) + g(x_{2L/3+1}) + h(x_L) + F(x_L, W_L),$$
$$x_{L+1} = f(y_L)$$



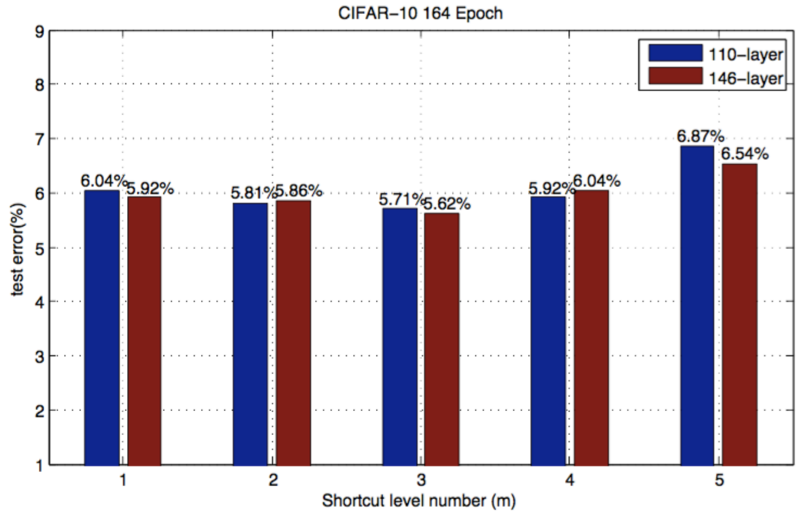


Fig. 4. Comparison of RoR with different shortcut level m. When m=1, it is the original ResNets. When m=3, RoR reaches the best performance.

Experiments

TABLE V
TEST ERROR (%) ON CIFAR-10/100 BY PRE-RESNETS AND PRE-ROR

500 Epoch	Pre-ResNets	Pre-RoR-3	Pre-ResNest+SD	Pre-RoR-3+SD
164-layer CIFAR-10	5.04	5.02	4.67	4.51
164-layer CIFAR-100	25.54	25.33	22.49	21.94

TABLE VI
TEST ERROR (%) ON CIFAR-10/100 BY WRN AND ROR-WRN

500 Epoch	WRN40-2	RoR-3-WRN40-2	WRN40-2+SD	RoR-3-WRN40-2+SD
CIFAR-10	4.81	5.01	4.80	4.59
CIFAR-100	24.70	25.19	22.87	22.48

TABLE XI
TEST ERROR (%) ON CIFAR-10, CIFAR-100 AND SVHN BY DIFFERENT METHODS

Method(#Parameters)	CIFAR-10	CIFAR-100	SVHN
NIN [5]	8.81	35.68	2.35
FitNet [8]	8.39	35.04	2.42
DSN [9]	7.97	34.57	1.92
All-CNN [10]	7.25	33.71	-
Highway [28]	7.72	32.39	-
ELU [22]	6.55	24.28	-
FractalNet (30M) [29]	4.59	22.85	1.87
ResNets-164 (2.5M) [12] (reported by [13])	5.93	25.16	-
FitResNet, LSUV [26]	5.84	27.66	-
Pre-ResNets-164 (2.5M) [13]	5.46	24.33	-
Pre-ResNets-1001 (10.2M) [13]	4.62	22.71	-
ELU-ResNets-110 (1.7M) [31]	5.62	26.55	-
PELU-ResNets-110 (1.7M) [24]	5.37	25.04	-
ResNets-110+SD (1.7M) [15]	5.23	24.58	1.75 (152-layer)
ResNet in ResNet (10.3M) [30]	5.01	22.90	-
SwapOut (7.4M) [32]	4.76	22.72	-
WResNet-d (19.3M) [33]	4.70	-	-
WRN28-10 (36.5M) [14]	4.17	20.50	1.64
CRMN-28 (more than 40M) [34]	4.65	20.35	1.68
RoR-3-164 (2.5M)	4.86	22.47	-
Pre-RoR-3-164 (2.5M)	4.51	21.94	-
RoR-3-WRN40-2 (2.2M)	4.59	22.48	-
Pre-RoR-3-1202 (19.4M)	4.49	20.64	-
RoR-3-WRN40-4 (8.9M)	4.09	20.11	-
RoR-3-WRN58-4 (13.3M)	**3.77**	**19.73**	**1.59**

Thank you