Resnet review

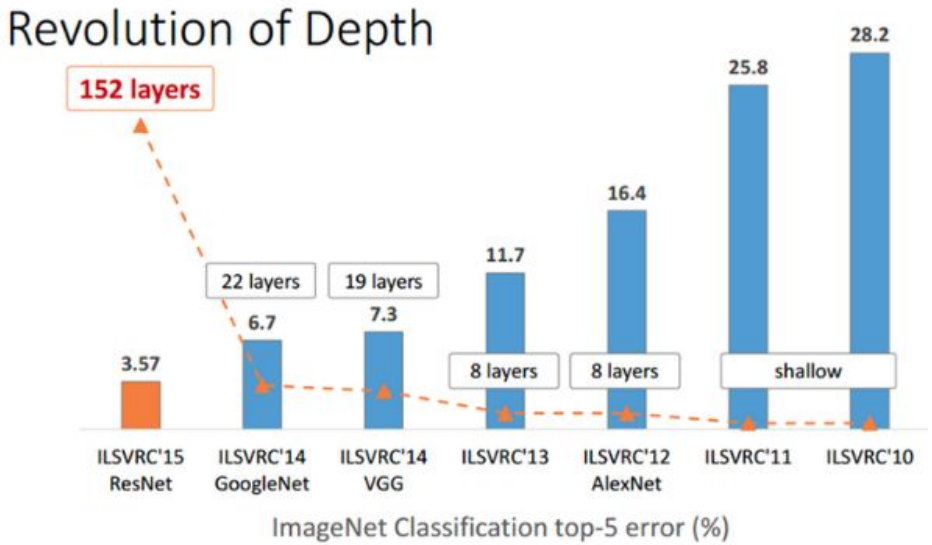
By beomgon.yu

Contents

1. Introduction
2. Reformulation by suboptimal problem(residual leaning)
3. Gradient pumping(by shortcut connection)
4. trainable Non linearity
5. etc

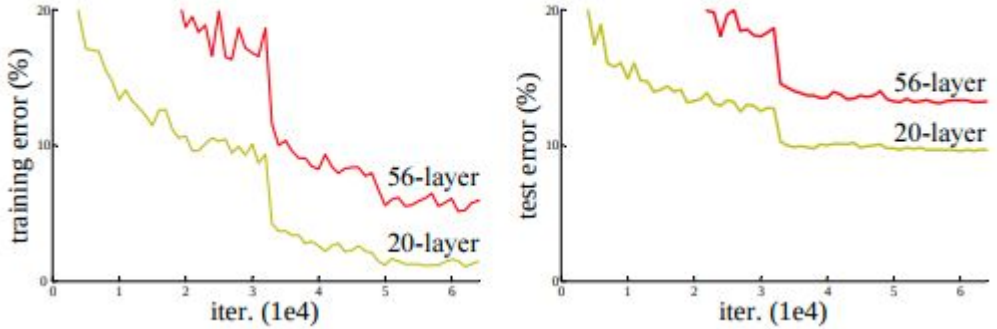
Introduction

Deeper network get better performance
Resnet use lots of layer, first time,
Is there a problem or how they could do?



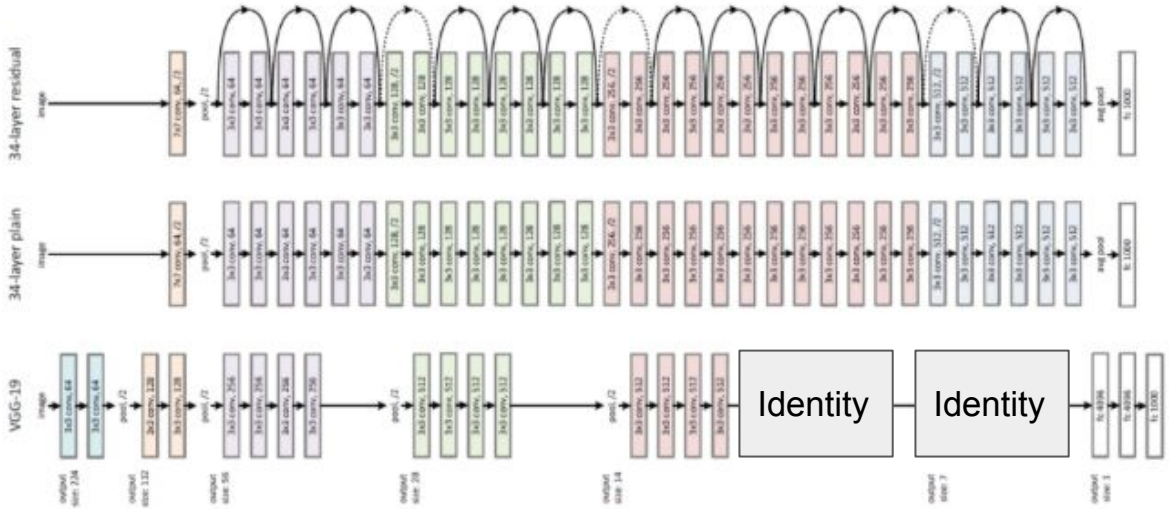
Introduction

Deeper network get better performance,
But at some point, **performance degradation** happened.
This is neither overfitting, nor vanishing gradient
(normalized initialization and batch normalization)



Introduction

Deeper network, Is there a No solution, NO?
Solution must be exist,(counter part shallow network)
But its difficult to learn Identity, or takes long time because of Non linearity

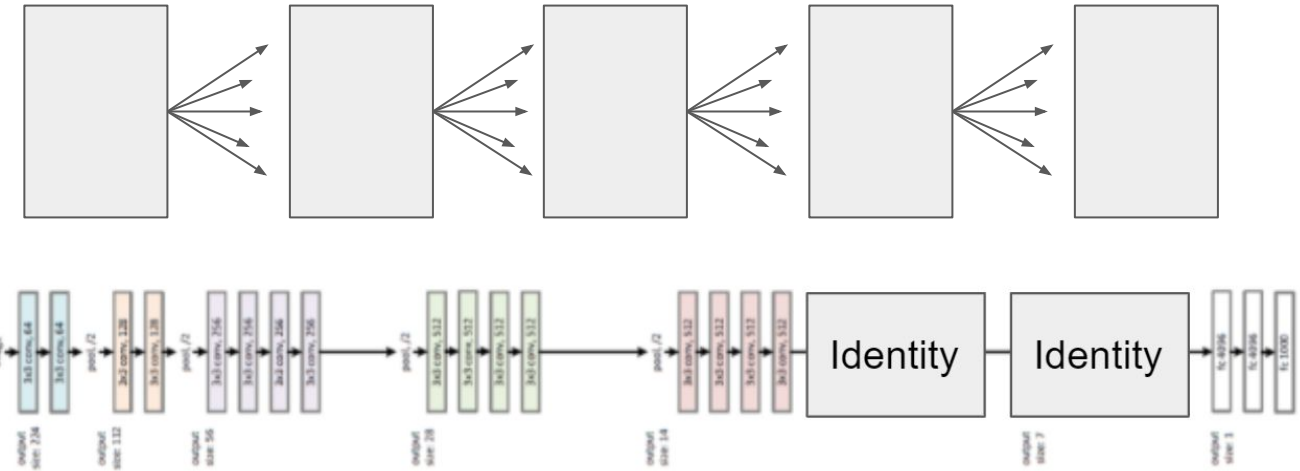


Reformulation by suboptimal problem

difficult optimization problem

divide by multiple sub optimization problem,

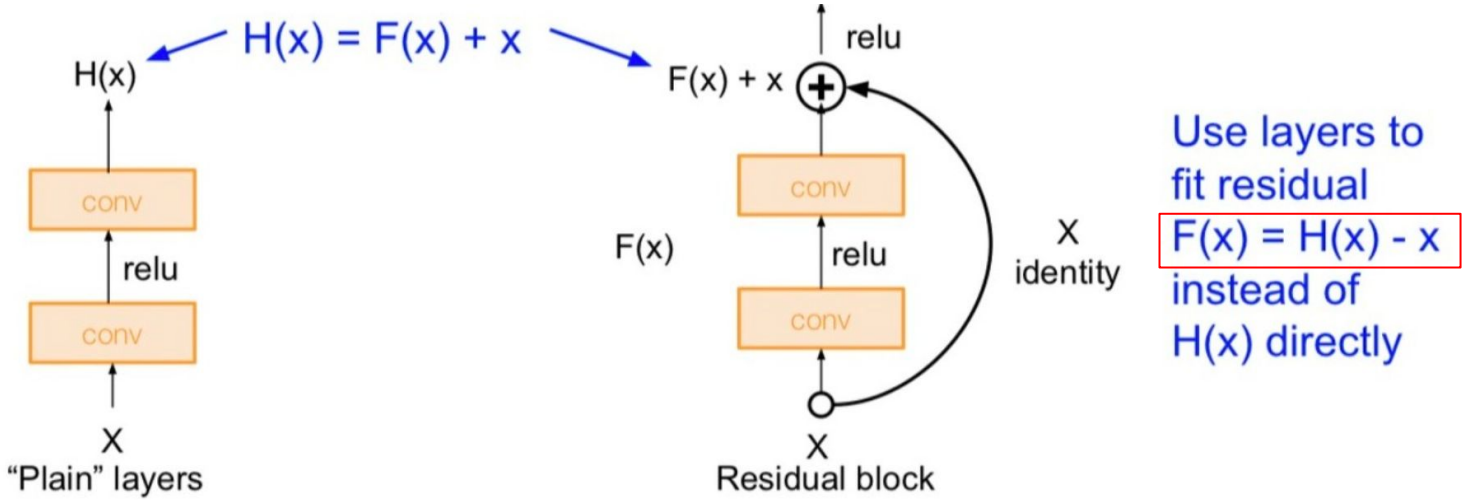
Not optimal solution but sub optimal solution



Reformulation by suboptimal problem

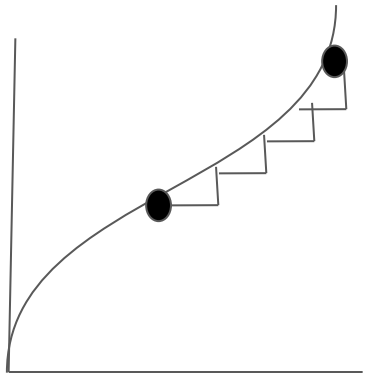
residual learning

instead of direct learning of mapping, just learning residual (offset)
(forced to learn identity)
(offset, its similar to anchor in object detection)
Because of non linearity, learning identity is difficult



Reformulation by suboptimal problem

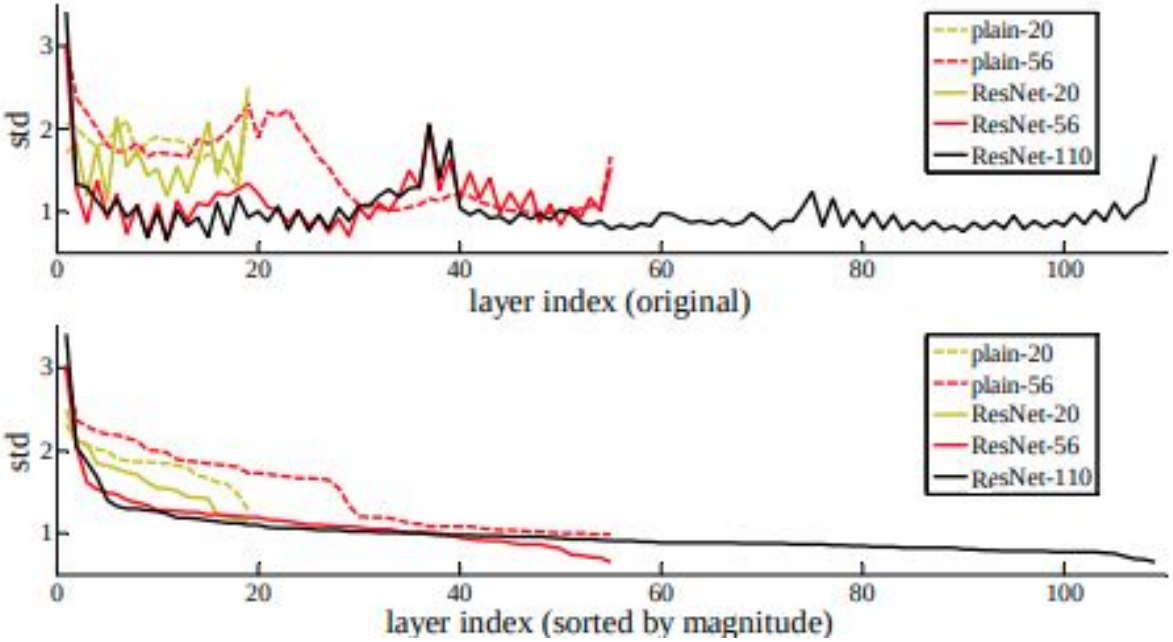
천리길도 한 걸음부터

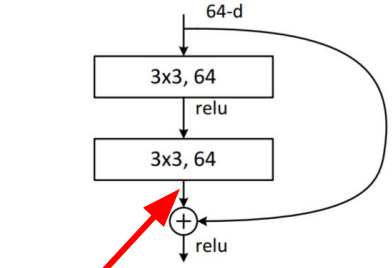


Reformulation by suboptimal problem

Analysis of Layer Responses

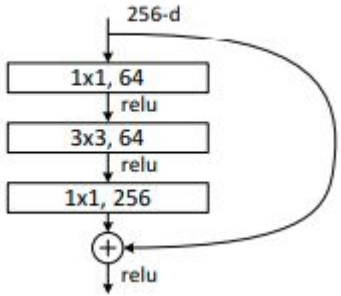
ResNets have generally smaller responses than their plain counterparts





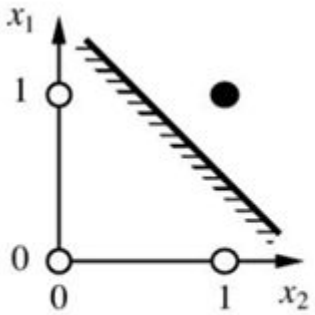
Gradient pumping

this optimization difficulty is unlikely to be caused by vanishing gradients
trained with BN, and checked each propagated gradient has healthy norms with BN
But maybe skip connection give gradient directly to lower layer, so help solving vanishing gradient problem or fast learning

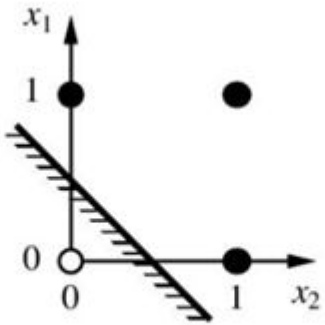


trainable Non linearity

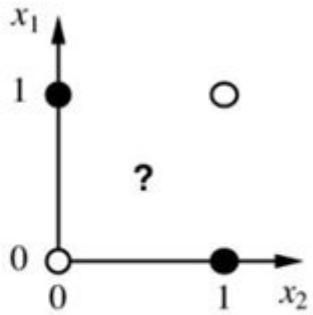
Importance of non linearity
without non linearity, mlp or higher nn can not solve trivial xor problem.



x_1 and x_2



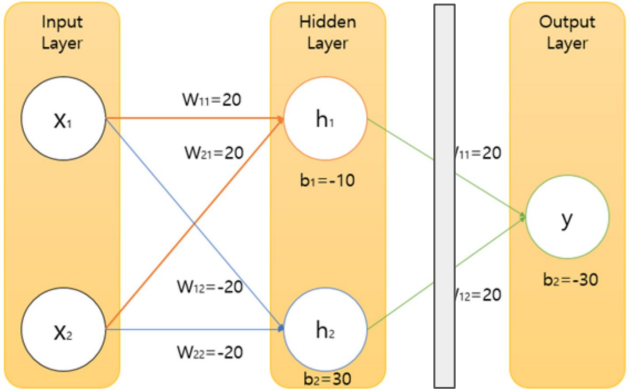
x_1 or x_2



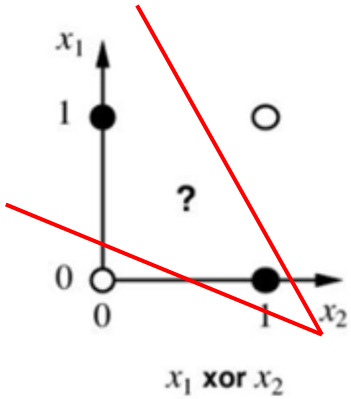
x_1 xor x_2

trainable Non linearity

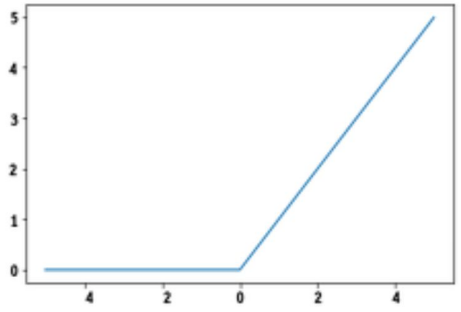
Importance of non linearity
with non linearity, xor can be solved.
how about using more complex non linearity or trainable non linearity?



Relu



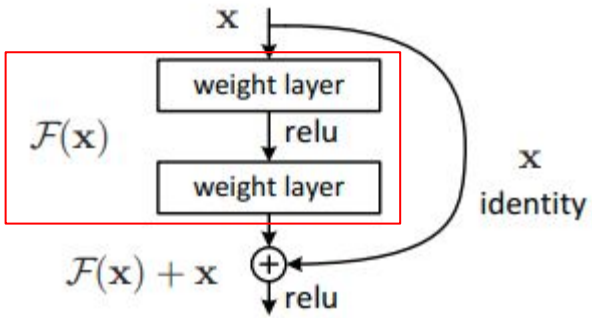
x1 xor x2

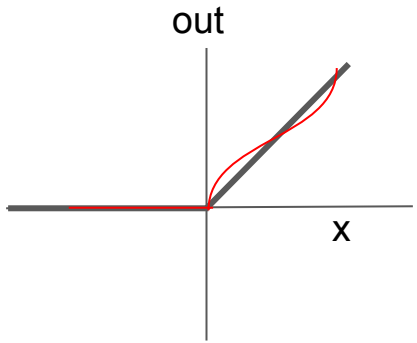


Relu

trainable Non linearity

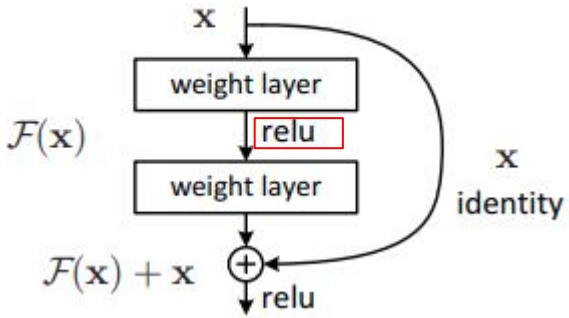
How about if non linearity can be trainable?

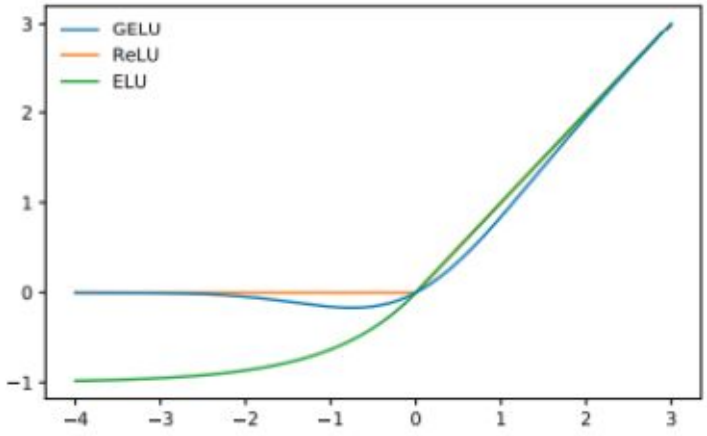




trainable Non linearity

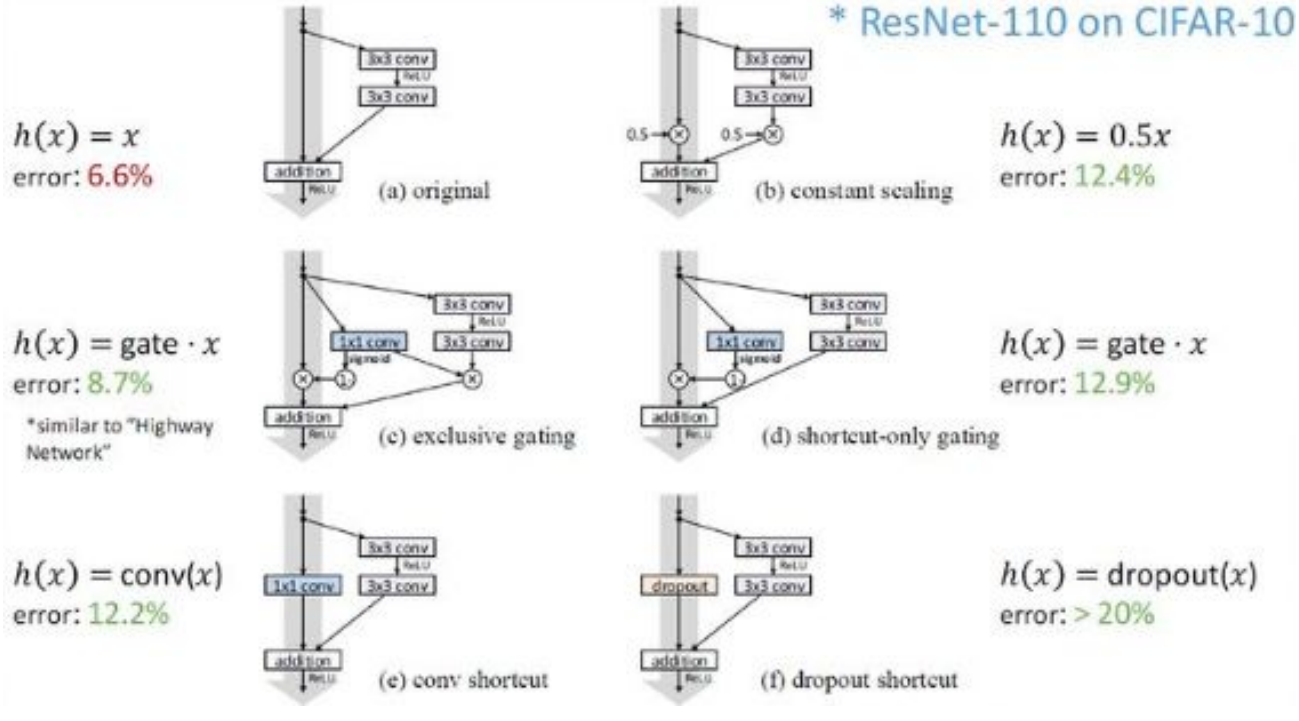
in resnet 18, with caltech 101 dataset,
gelu >= relu, leaky relu, elu > sigmoid, tanh



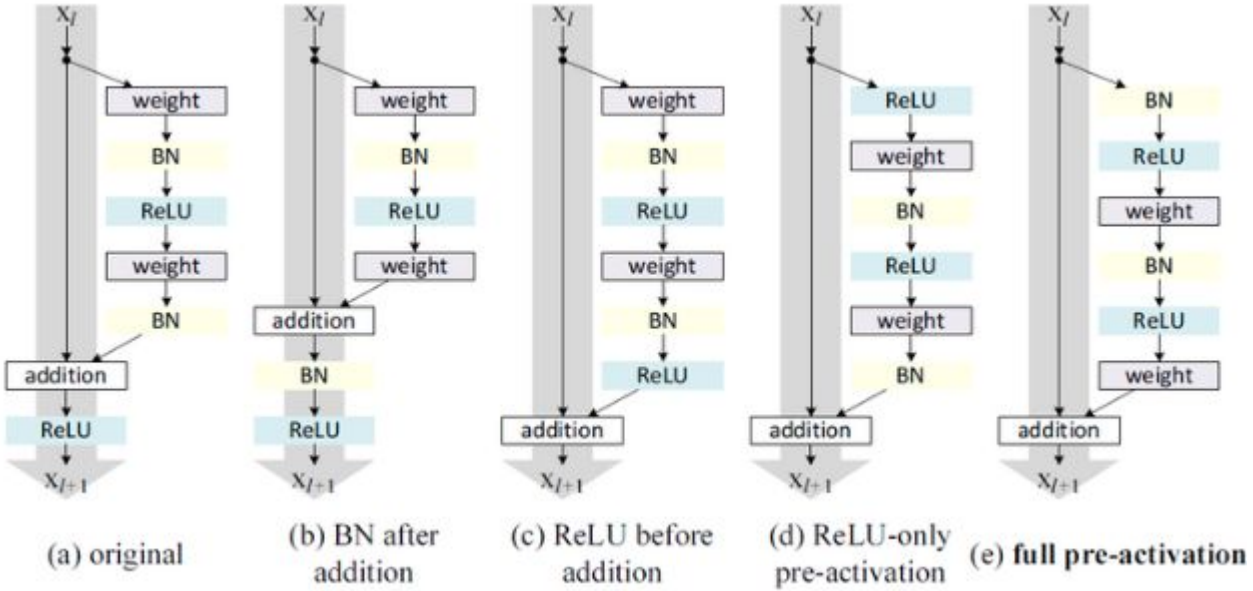


ETC

skip connection variation



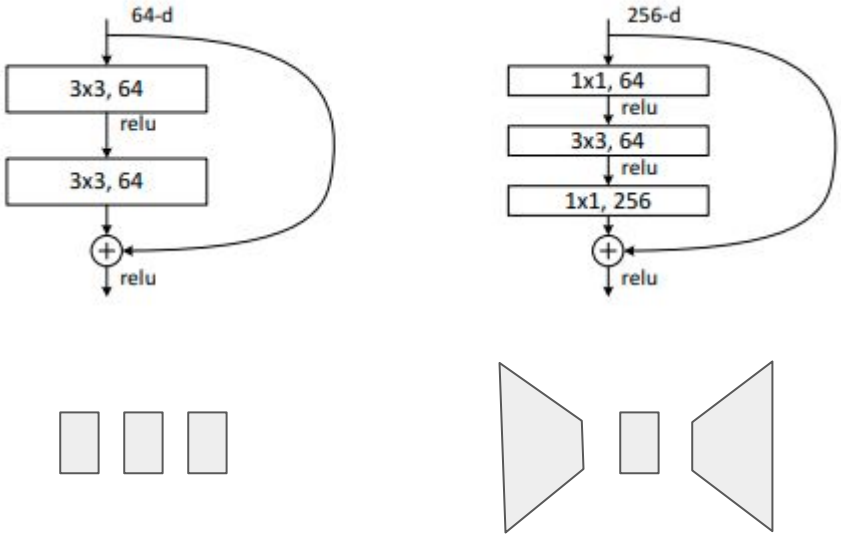
ETC


x_l
weight
BN
ReLU
weight
BN
addition
ReLU
x_{l+1}
(a) original
(b) BN after addition
(c) ReLU before addition
(d) ReLU-only pre-activation
(e) full pre-activation

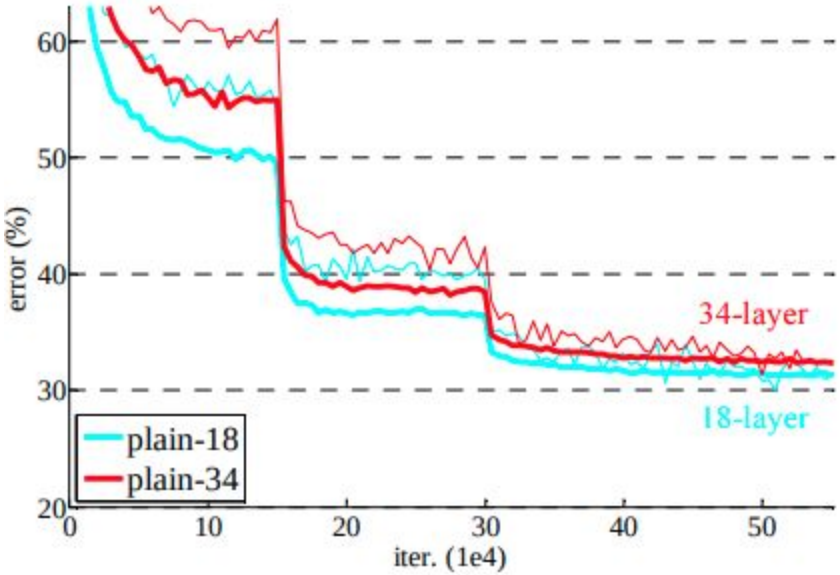
ETC

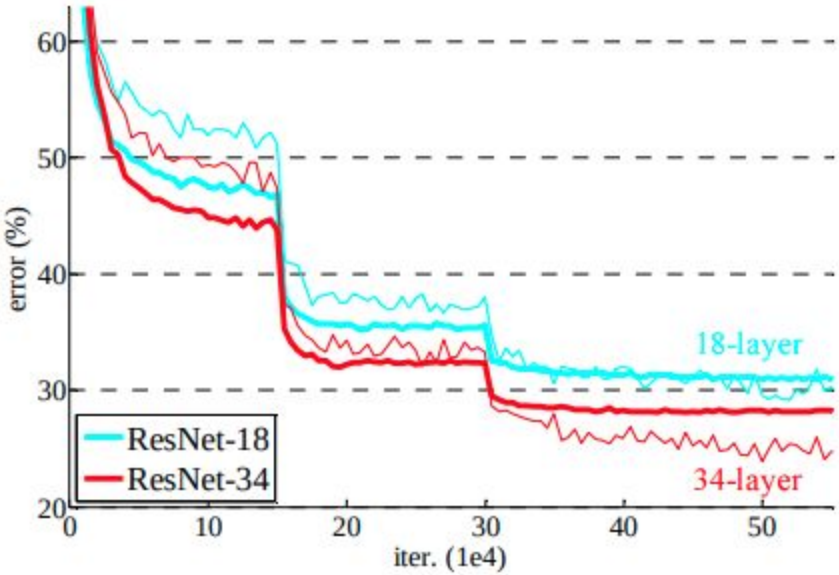
bottleneck layer

for computation efficiency, use it for more than 50 layers.



Test Result


60
50
40
30
20
error (%)
0
10
20
30
40
50
iter. (1e4)
plain-18
plain-34
34-layer
18-layer


60
50
40
30
20
error (%)
0
10
20
30
40
50
iter. (1e4)
ResNet-18
ResNet-34
18-layer
34-layer

Thank you !!!