# Deep Residual Networks

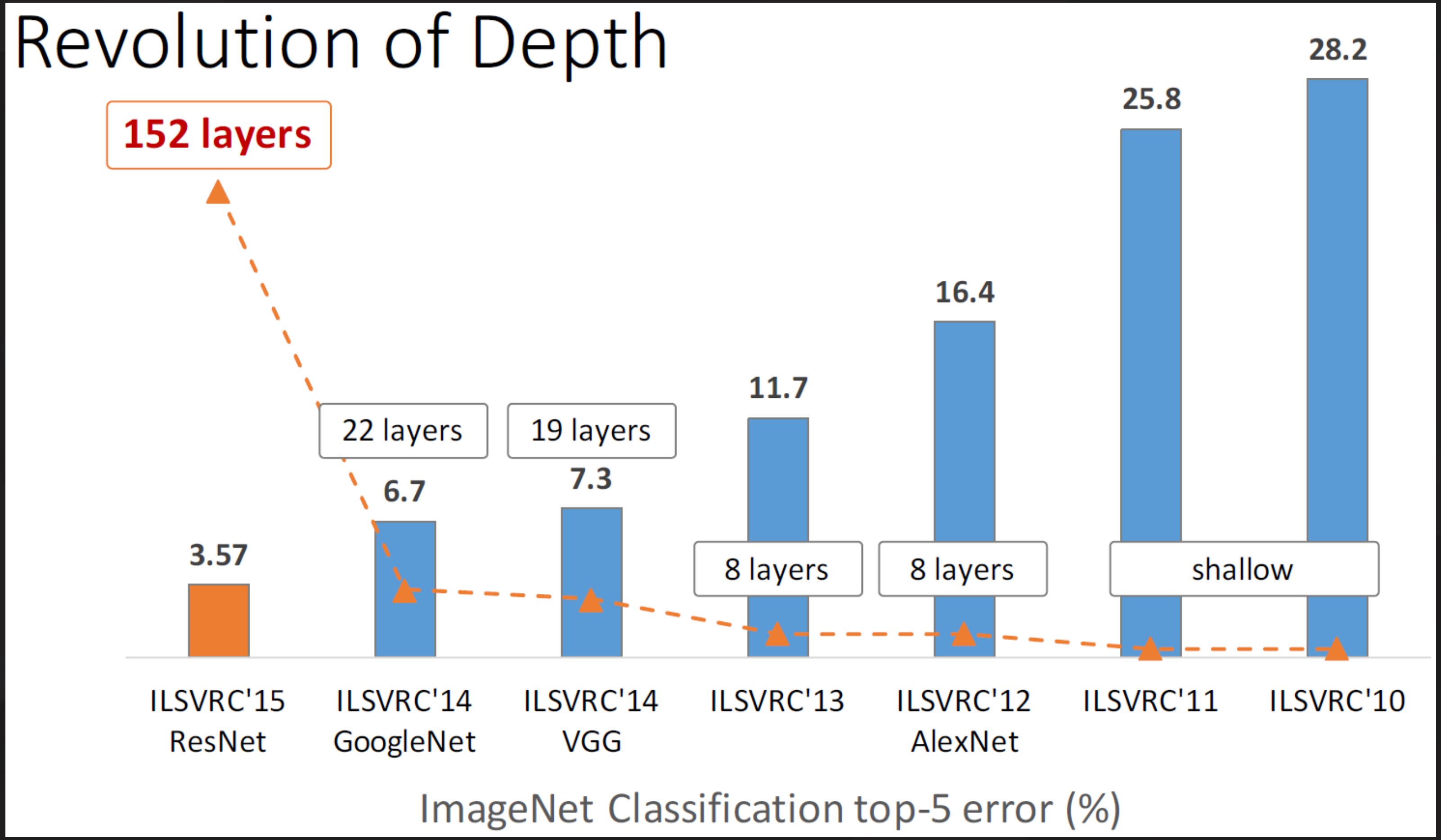
Reporter ： Liyan Sun

Kaiming He et al, “Deep Residual Learning for Image Recongnition” ,
CVPR 2016 (oral & best paper award), Google scholar citation: 468

Kaiming He et al, “Identity Mappings in Deep Residual Networks” ,
ECCV 2016 (spotlight), Google scholar citation: 34

Andreas Veit et al, “Residual Networks are Exponential Ensembles of Relatively Shallow Networks” ,
NIPS 2016, Google scholar citation: 2

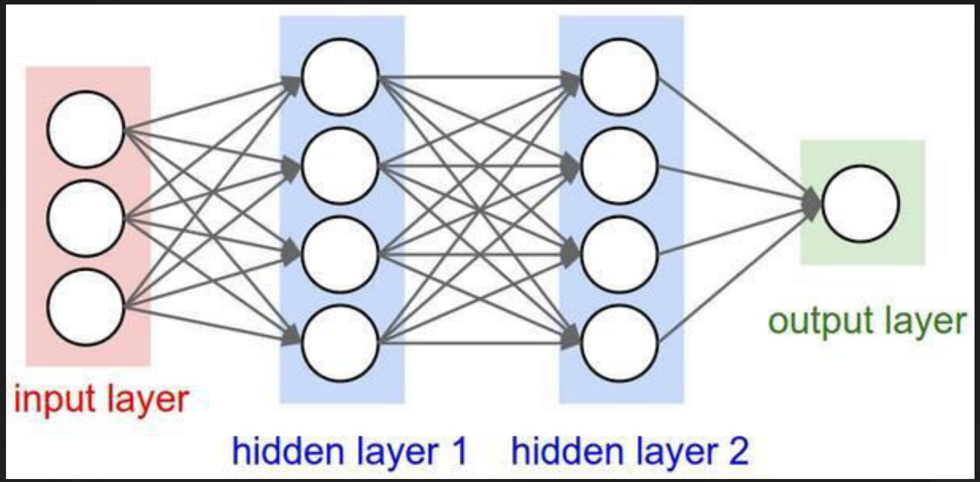
# Evolution of deep networks

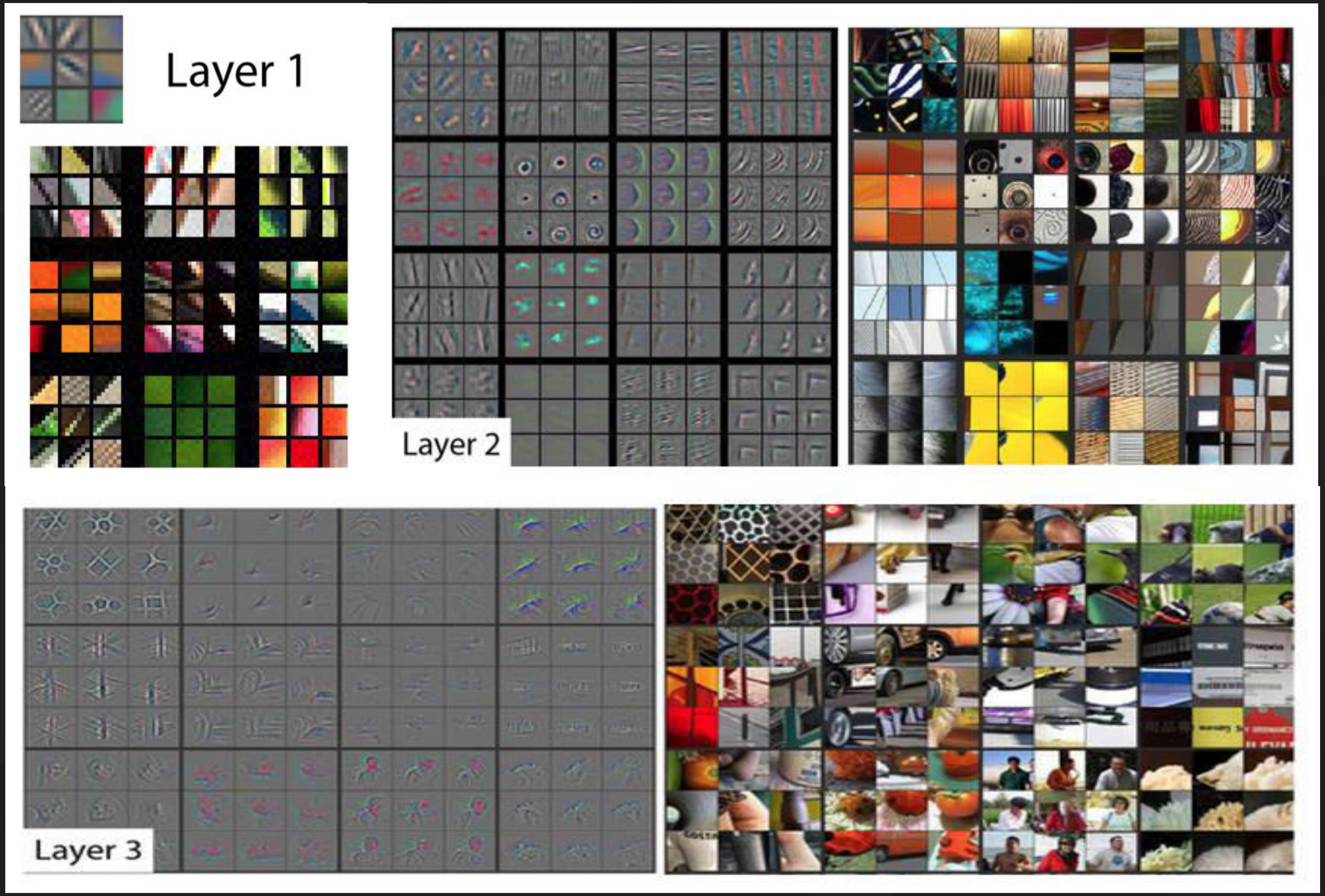


# What does depth mean?

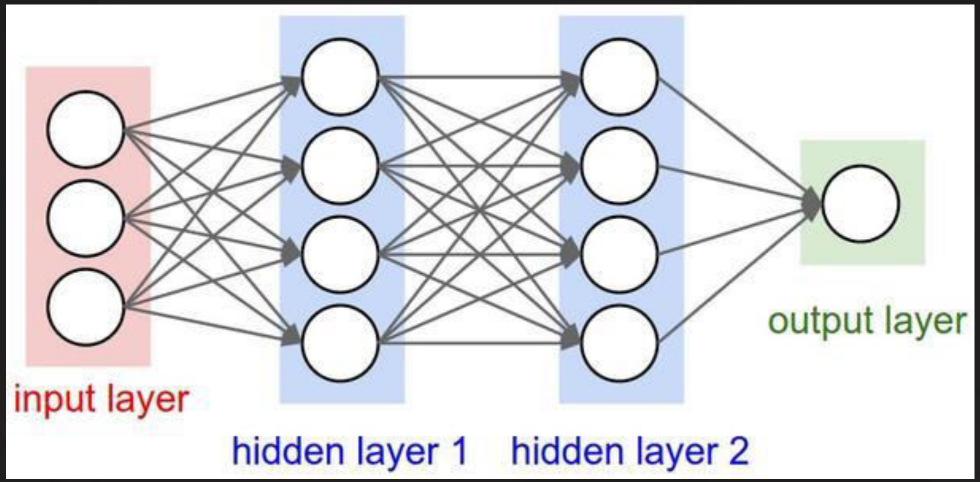
Deep ↑ Representation ability ↑

→ Forward (data flow)




Layer 1
Layer 2
Layer 3

# What does depth mean?




Backward (gradient flow)

Is optimization is as easy as stacking layers?

# Gradients Vanishing

- The multiplying property of gradients causes the phenomenon.
- This can be addressed by:

  ①Normalized Initialization

  ②Batch Normalization

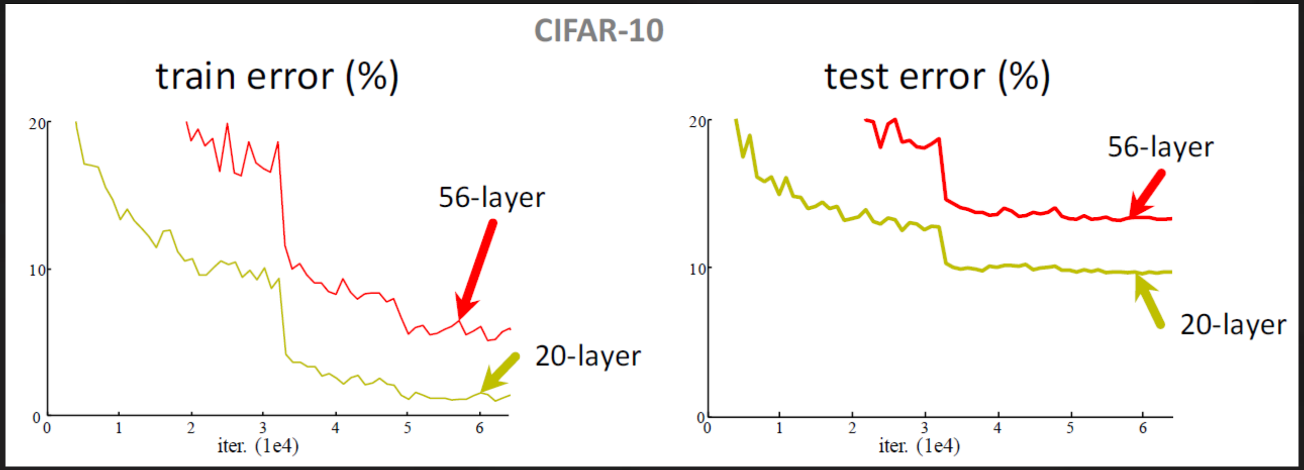
$$\rightarrow \boxed{\text{layer}} \rightarrow x \rightarrow \quad \hat{x} = \frac{x-\mu}{\sigma} \quad \rightarrow \quad y = \gamma\hat{x} + \beta$$

  ③Appropriate activation function

  sigmoid(x) → ReLU(x)

# Performance Saturation/degradation

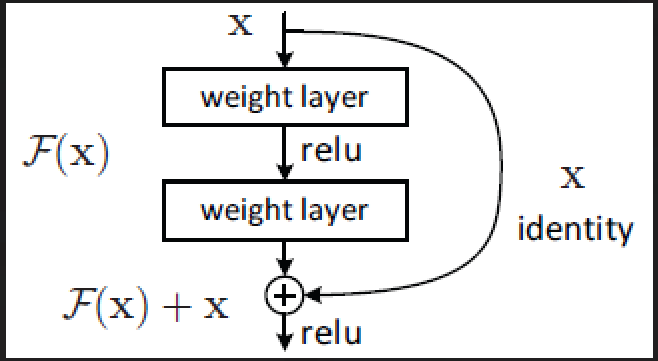
- “plain” networks on CIFAR-10




① Rich solution space
② deeper “should” means lower training error:
(1)Original layers: copies of shallow ones
(2)Extra layers: set as identities
(3)Results: same training error
③ Networks cannot find solutions when going deeper

# Residual Learning Block

- Define H(x) = F(x) + x, the stacked weight layers try to approximate F(x) instead of H(x).

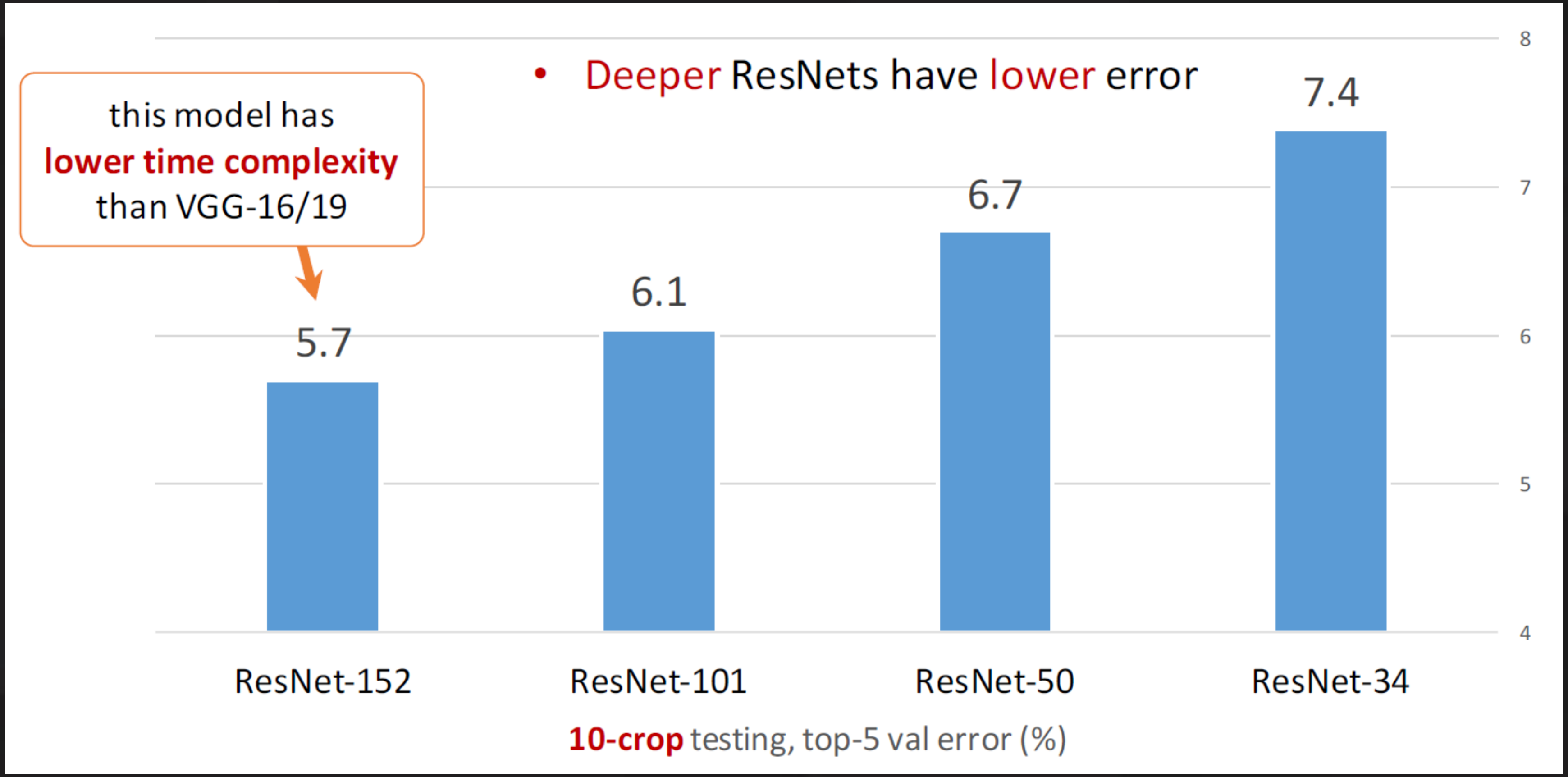



If the optimal function is close to identity mapping, the nonlinear stacked weight layers can capture the small perturbations easier.

① No extra parameter and computation complexity introduced.

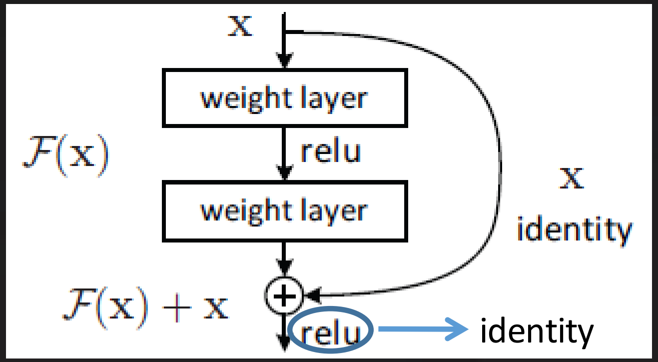
② Element-wise addition is performed on all feature maps.

# ResNet can be deeper



# The Insight of Identity mapping

- We turn the ReLU activation function after the addition into a identity mapping.




$$x_{l+1} = x_l + F(x_l)$$

$$x_{l+2} = x_{l+1} + F(x_{l+1})$$

$$x_{l+2} = x_l + F(x_l) + F(x_{l+1})$$

$$\downarrow$$

$$x_L = x_l + \sum_{i=l}^{L-1} F(x_i)$$

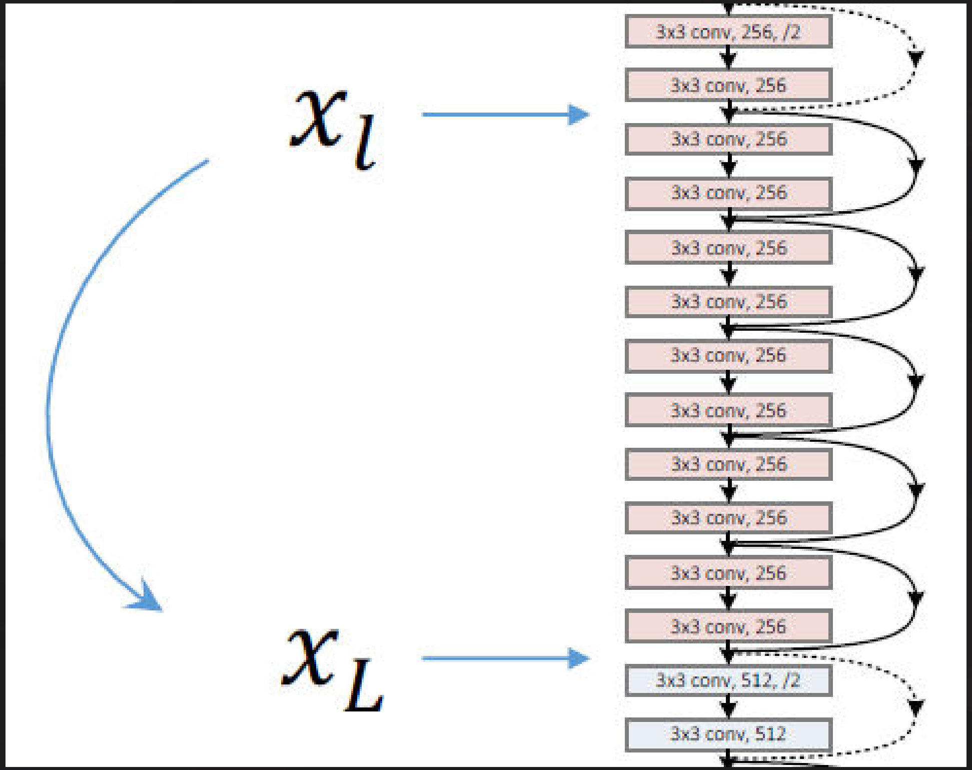
# Smooth Forward Propagation

- Any $x_l$ is directly forward-prop to any $x_L$, plus residual.
- Any $x_l$ is additive outcome.

  In contrast to the multiplicity:

$$x_L = x_l + \sum_{i=l}^{L-1} F(x_i)$$

$$x_L = \prod_{i=l}^{L-1} W_i x_l$$



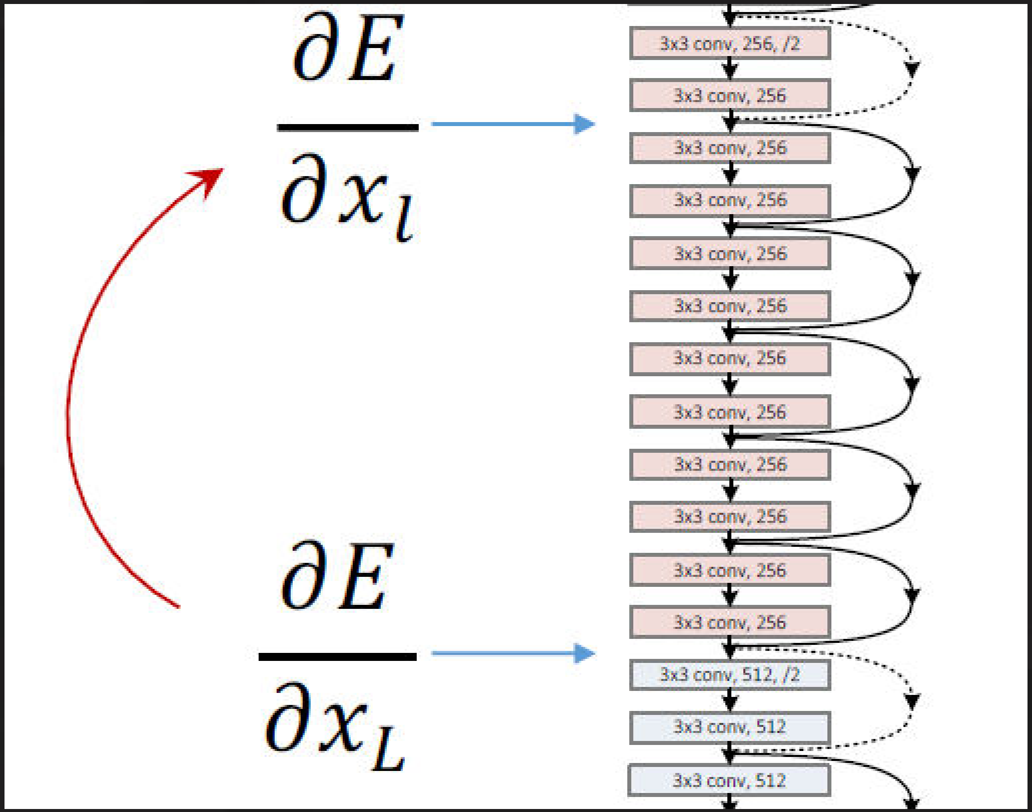
# Smooth Backward Propagation

- The gradients flow is also in the form of addition.
- The gradients of any layer is unlikely to vanish.

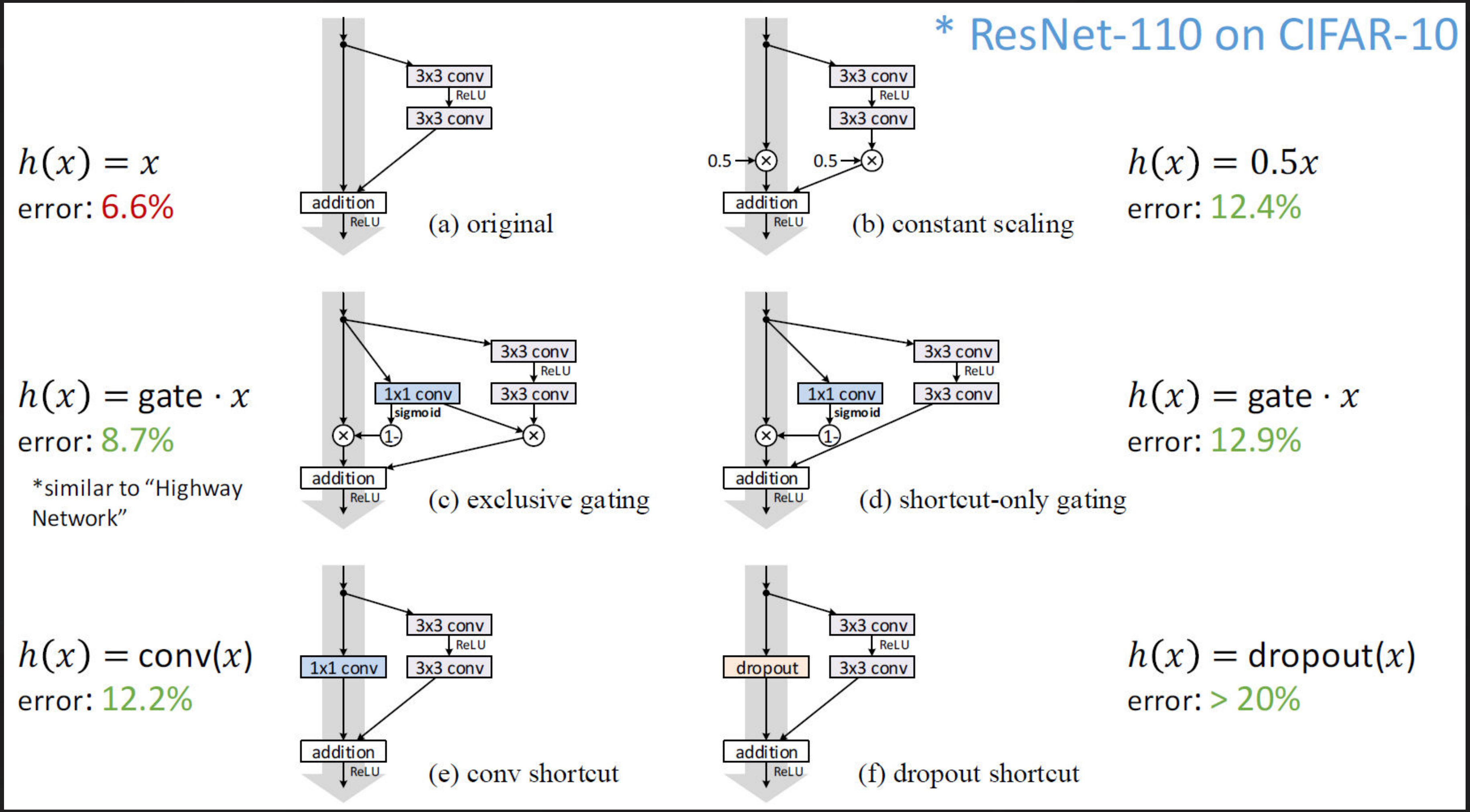
  In contrast to the multiplicity

$$x_L = x_l + \sum_{i=l}^{L-1} F(x_i) \rightarrow \frac{\partial E}{\partial x_l} = \frac{\partial E}{\partial x_L}\frac{\partial x_L}{\partial x_l} = \frac{\partial E}{\partial x_L}\left(1 + \frac{\partial}{\partial x_l}\sum_{i=1}^{L-1} F(x_i)\right)$$

$$x_L = \prod_{i=l}^{L-1} W_i x_l \rightarrow \frac{\partial E}{\partial x_l} = \prod_{i=l}^{L-1} W_i \frac{\partial E}{\partial x_L}$$



# What if shortcut mapping h(x) ≠ identity?



# If scaling the shortcut

- If h is multiplicative, e.g. $h(x) = \lambda x$, the forward and backward is denoted as

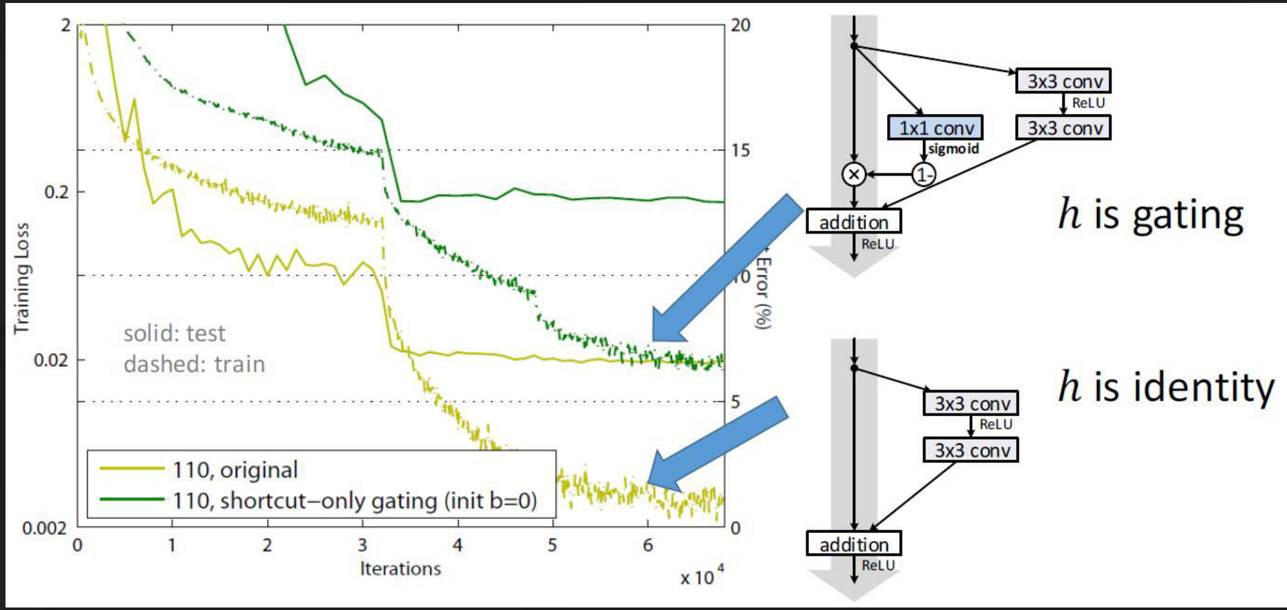
$$x_L = \lambda^{L-l} x_l + \sum_{i=l}^{L-1} \hat{F}(x_i)$$

$$\frac{\partial E}{\partial x_l} = \frac{\partial E}{\partial x_L}\left(\lambda^{L-l} + \frac{\partial}{\partial x_l}\sum_{i=1}^{L-1} \hat{F}(x_i)\right)$$

- Either $\lambda$ is larger or smaller than 1 is problematic!

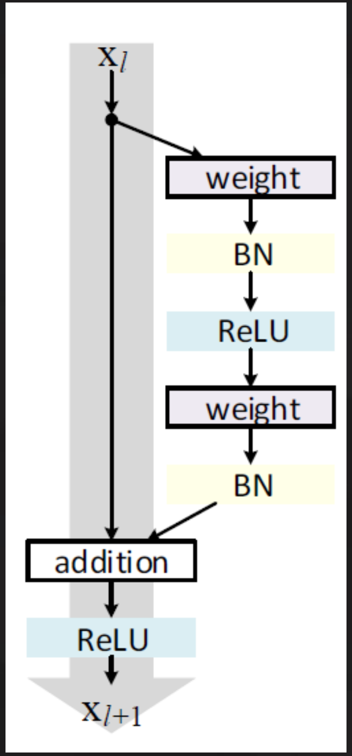
# If gating the shortcut

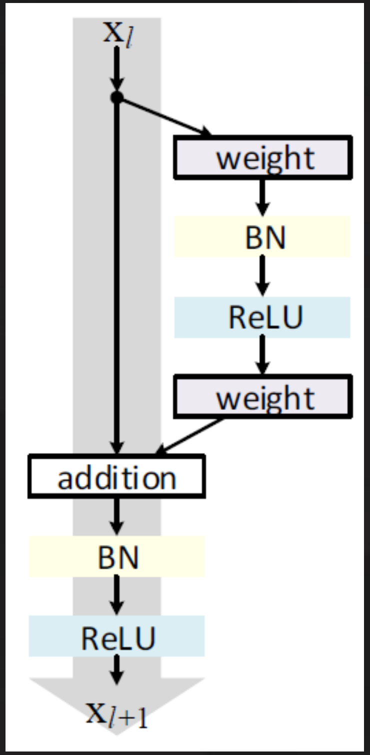
- The gating should increase the representation ability.
- It’s the optimization rather than the representation dominates the results!

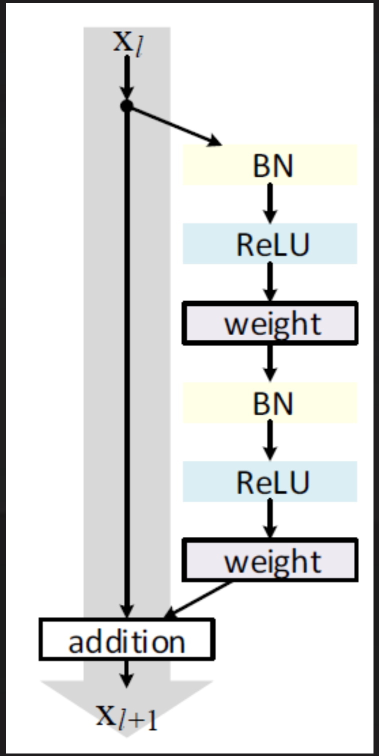


# If after-adding f(x) is identity mapping

- f(x) are tested in the below forms:
- f(x)=ReLU (original)
- f(x)=BN+ReLU
- f(x)=identity (pre-activation ResNet)

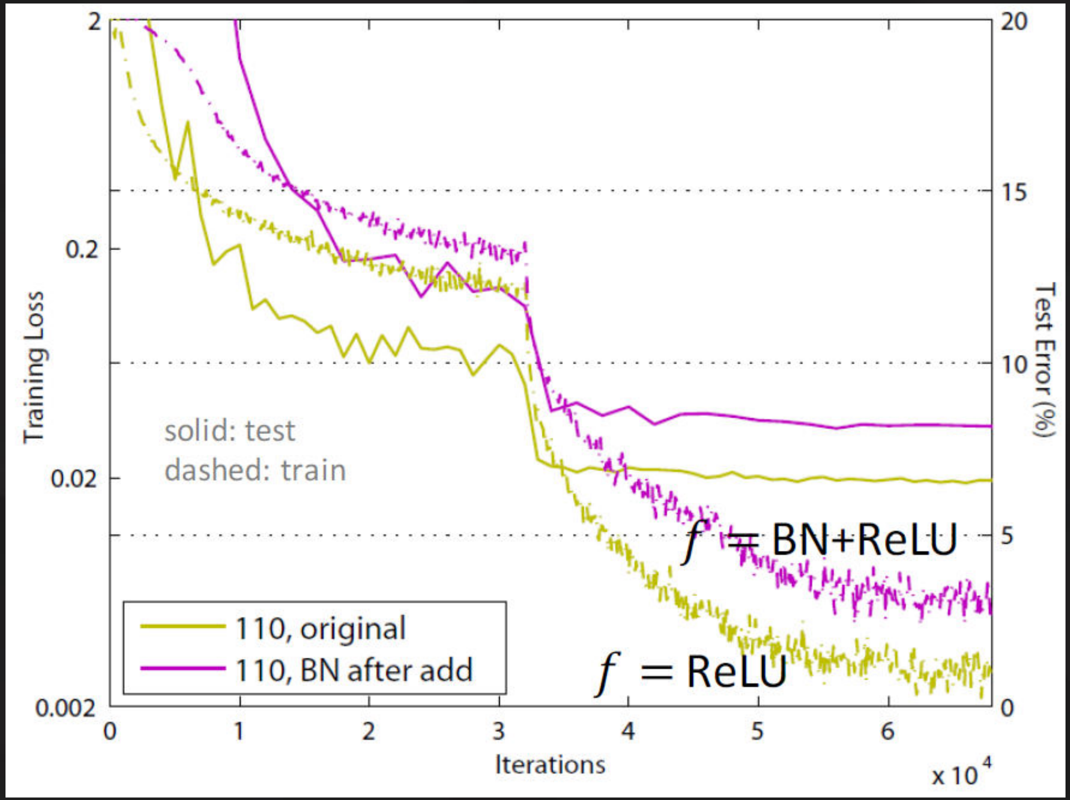


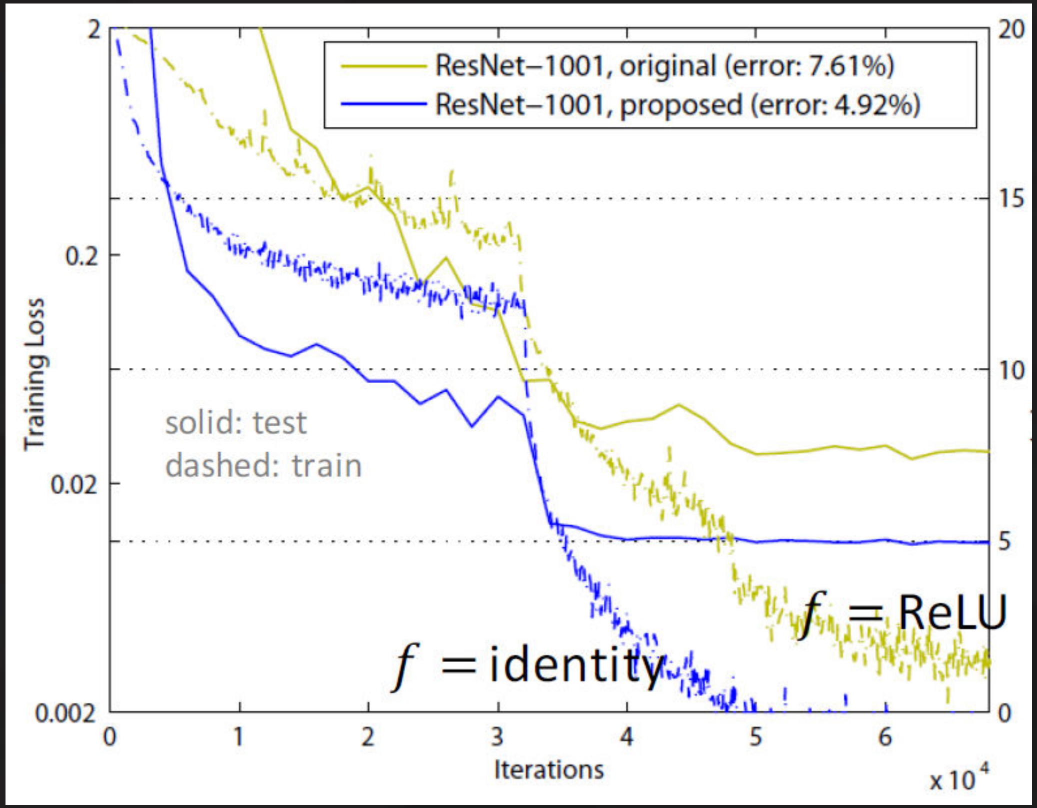
# ReLU v.s. ReLU+BN

- BN could block propagation.
- Keep the shortest path as smooth as possible.



# ReLU v.s. Identity

- ReLU could block propagation when the network is deep.
- Pre-activation ease the difficulty in optimization.



# ImageNet Results

ImageNet single-crop (320x320) val error

| method | data augmentation | top-1 error (%) | top-5 error (%) |
|---|---|---|---|
| ResNet-152, original | scale | 21.3 | 5.5 |
| ResNet-152, pre-activation | scale | 21.1 | 5.5 |
| ResNet-200, original | scale | 21.8 | 6.0 |
| ResNet-200, pre-activation | scale | **20.7** | **5.3** |
| ResNet-200, pre-activation | scale + aspect ratio | **20.1*** | **4.8*** |

# Conclusions from He

- Keep the shortest path as smooth(clean) as possible!

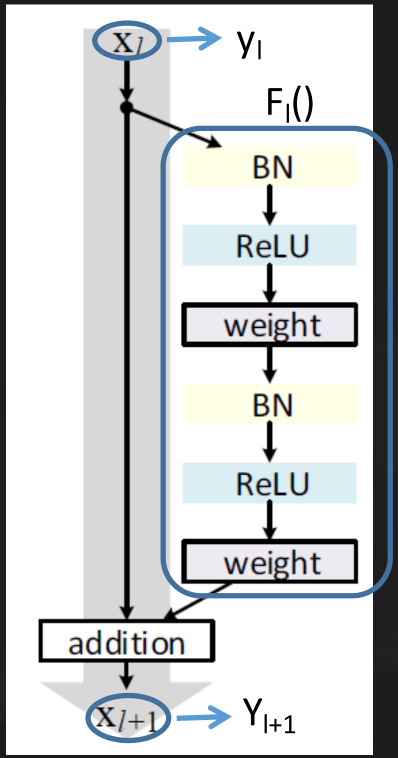
  By making h(x) and f(x) identity mapping.

  Forward and backward signals directly flow this path.

- Features of any layer is additive outcomes.

- 1000-layer ResNet can be easily trained and have better accuracy.

# Further expansion of the ResNet block




- According to previous analysis, and we replace $x_l$ with $y_l$ and F with $f_l$

$$x_L = x_l + \sum_{i=l}^{L-1} F(x_i)$$

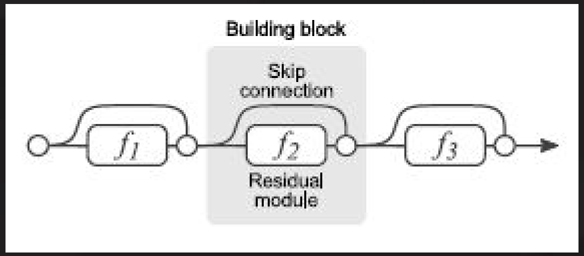
$$y_L = y_l + \sum_{i=l}^{L-1} f_l(y_l)$$

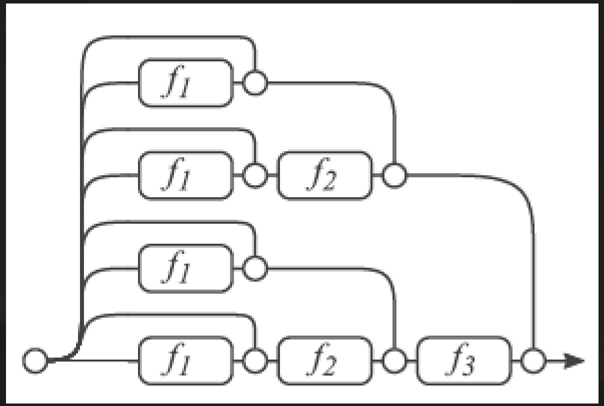
- We further expand this expression by unrolling the recursion in terms of basic input $y_l$.

# Example of the unrolling

- We take L=3 and l=0 for example for unrolling.
- The data flows along paths exponentially from input to output.
- We infer that residual networks have $2^n$ paths (multiplicity).

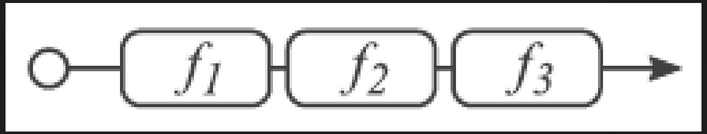
$$
\begin{aligned}
y_3 &= y_2 + f_3(y_2) \\
&= [y_1 + f_2(y_1)] \quad + \quad f_3(y_1 + f_2(y_1)) \\
&= \left[y_0 + f_1(y_0) + f_2(y_0 + f_1(y_0))\right] \; + \; f_3\left(y_0 + f_1(y_0) + f_2(y_0 + f_1(y_0))\right)
\end{aligned}
$$



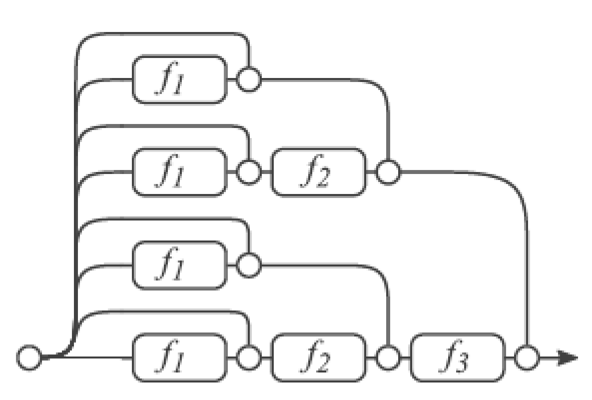



# Different from traditional NN

- In traditional NN, each layer depends only on the previous layer.




- In ResNet, each module $f_i()$ is fed data from a mixture of $2^{i-1}$ configuration of every possible combination of the previous i-1 residual modules.



# Theoretical Hypothesis

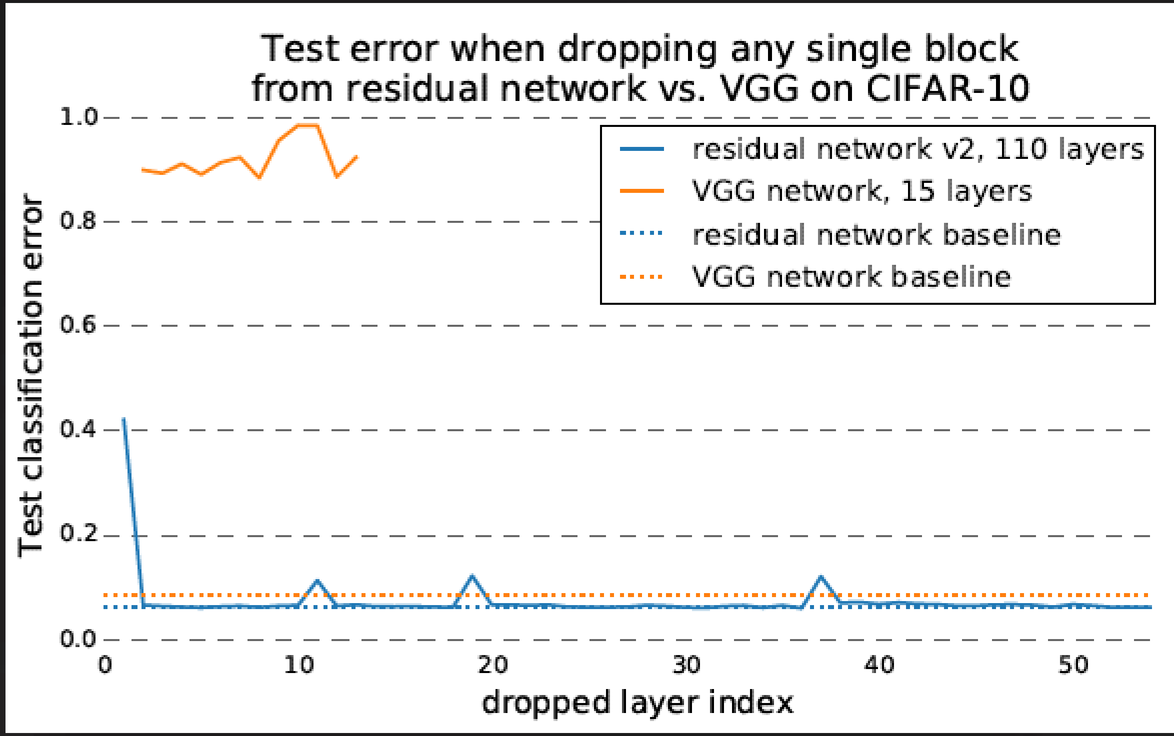
- Residual networks are not single ultra-deep networks, but very large implicit ensembles of many networks.

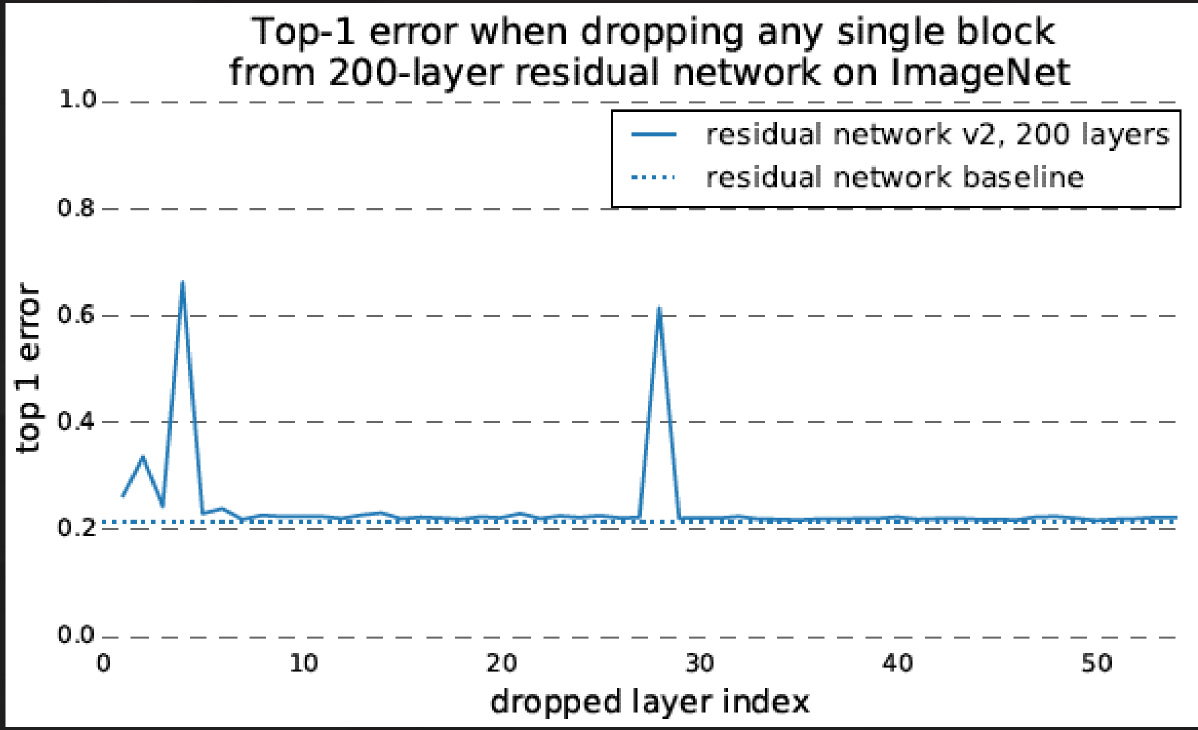
- This means depth may not be the only key idea in deep learning.

# Lesion Study

- Experiment 1: Deleting individual layers from neural networks.
- Experiment 2: Deleting many modules from residual networks.
- Experiment 3: Reordering modules in residual networks.

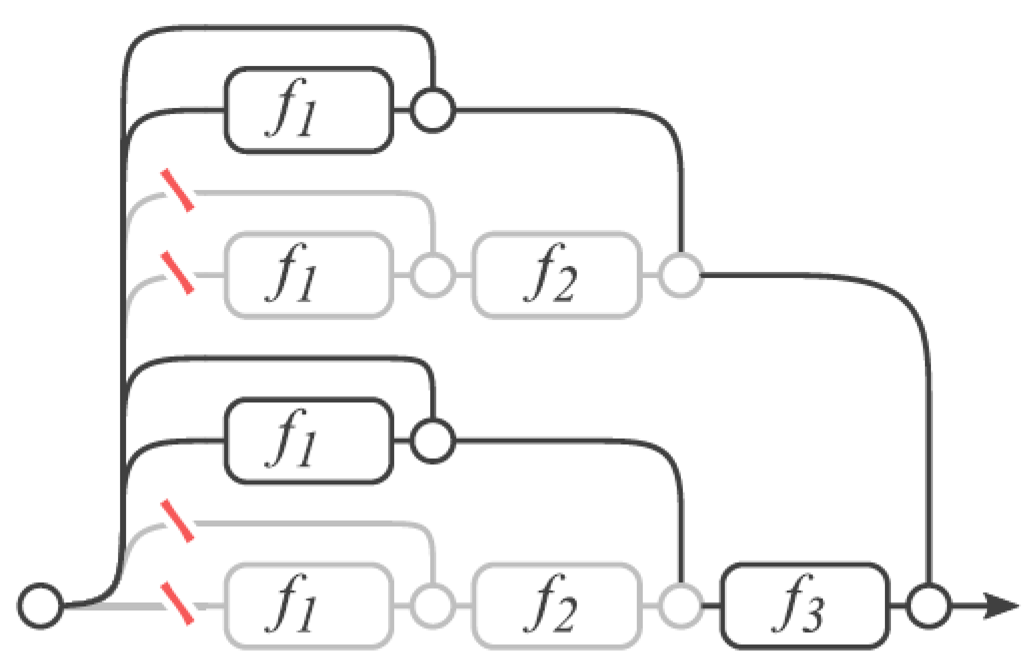
# Deleting individual modules in ResNet



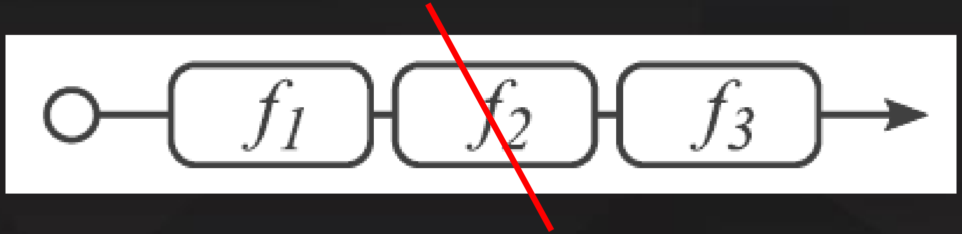
# For ResNet

- We show not all transformation within a residual network are necessary.
- It shows the importance of each building block.
- When a layer is removed, the effective number of paths is reduced from $2^n$ to $2^{n-1}$, leaving half of them valid.



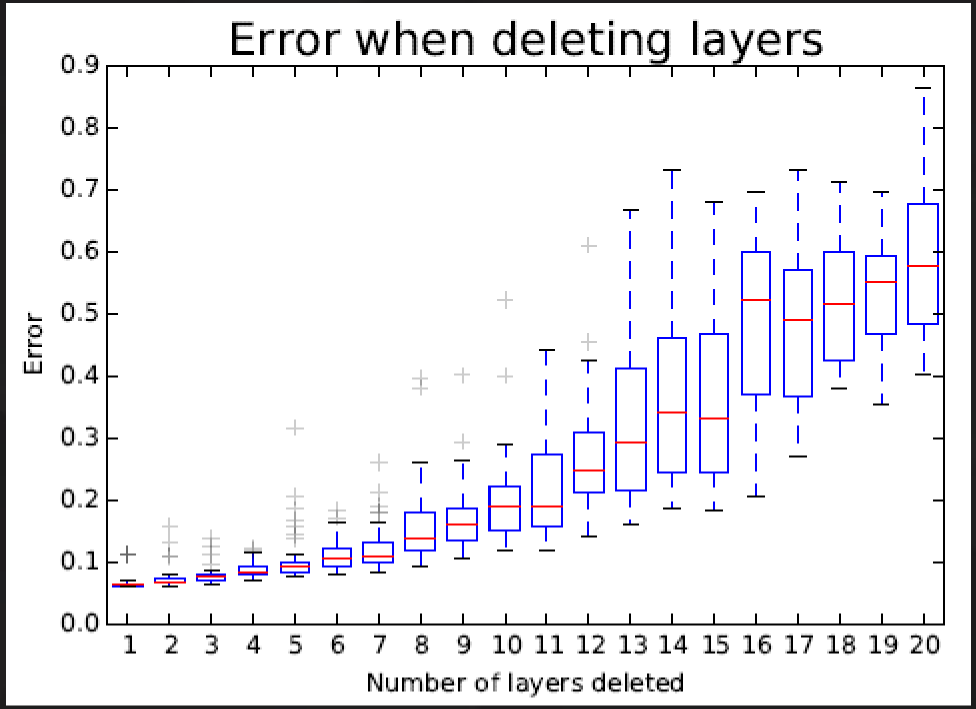
# For 12-layer VGG

- Deleting any layer in VGG reduces performance to chance level, because when a single layer is removed, the only viable path is corrupted.



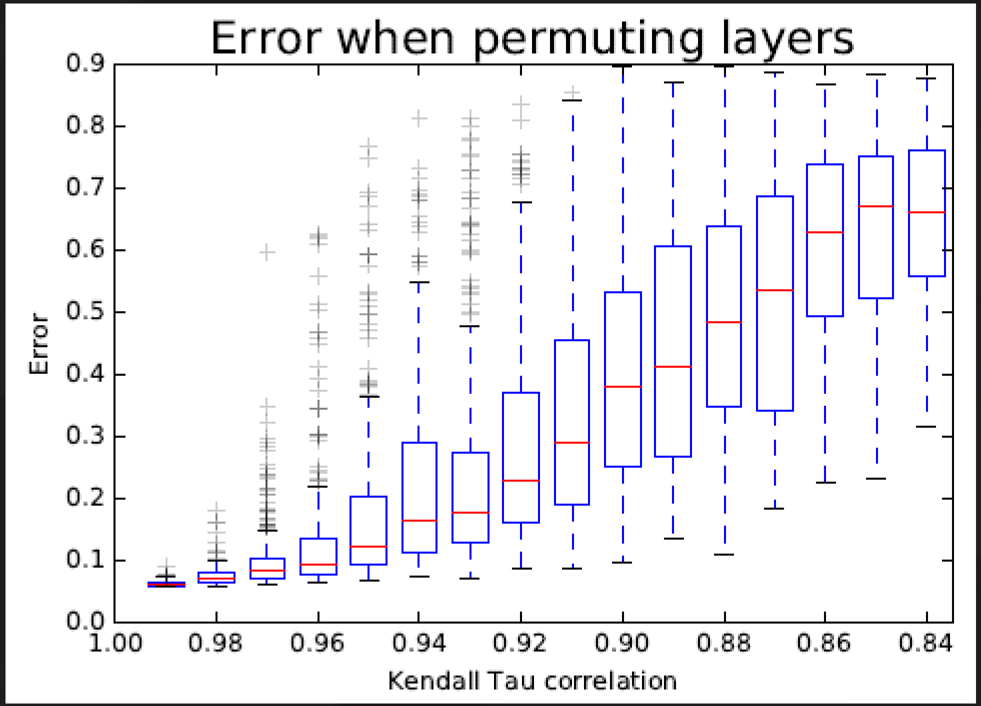
# Deleting many modules from ResNet

- One characteristic of ensembles is their performance depends smoothly on the number of members.
- When k residual modules are removed, the effective number of paths is reduced from $2^n$ to $2^{n-k}$.



# Reordering modules in ResNet

- We change the structure of ResNet by re-ordering the building blocks.
- We swap k randomly sampled pairs of building blocks.
- Kendall Tau correlation is adopted to measure the amount of corruption.



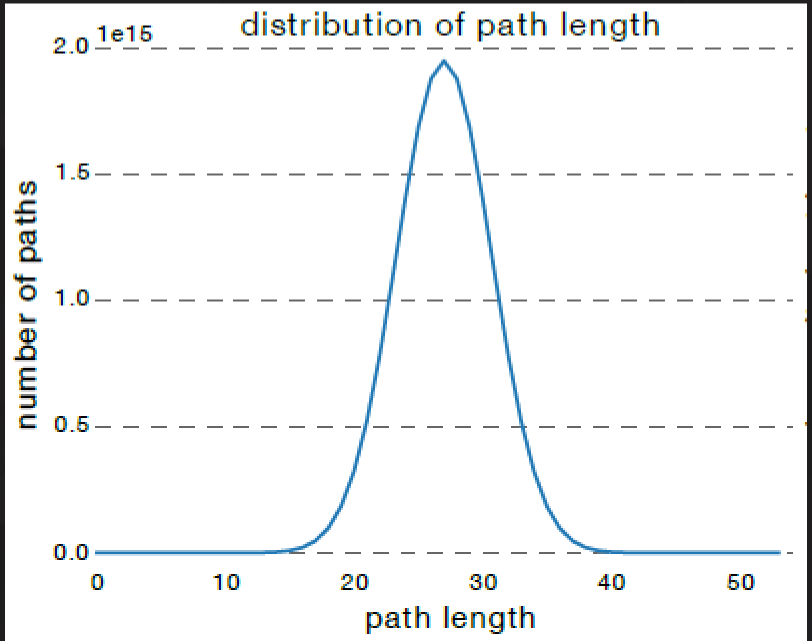
# The ensembles of relatively shallow networks

- Distribution of path lengths
- Vanishing gradients in residual networks
- Residual networks are exponential ensembles of relatively shallow networks

# Distribution of path lengths

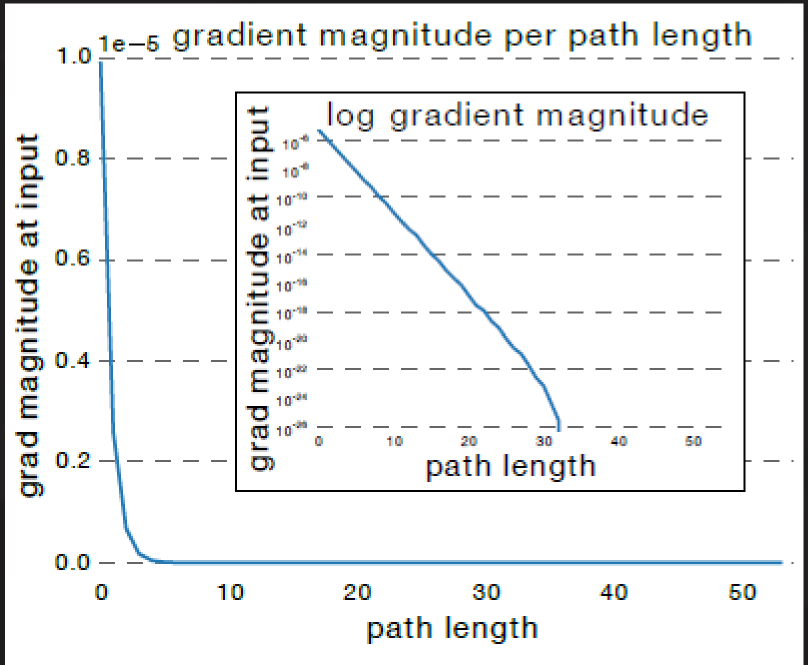
- Not all paths are of the same length.
- The distribution of all possible path lengths through the ResNet follows a Binomial distribution. The length of paths center around the mean of n/2.

$$p(l=k)=C_n^k p(1-p)$$



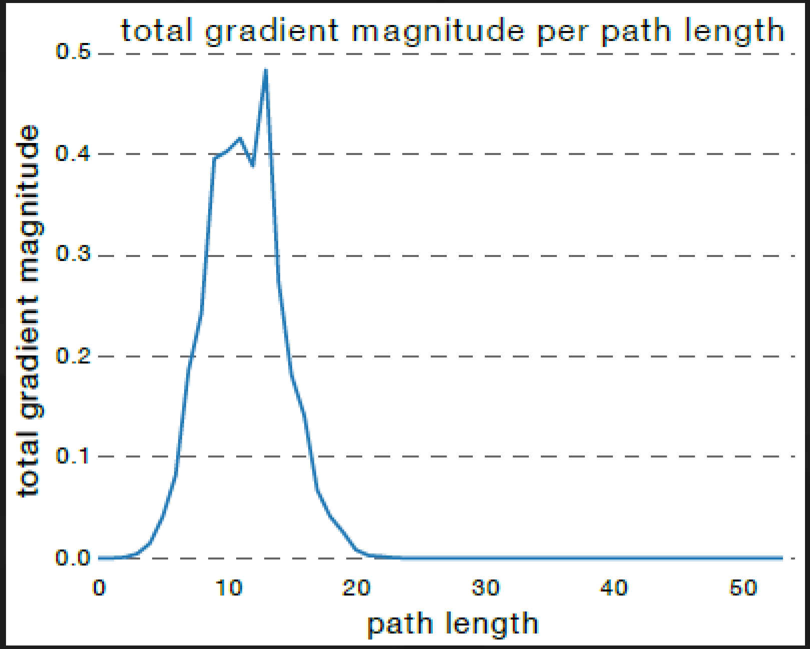
# Vanishing gradients in ResNet

- Data flows along all the paths in ResNet, while not all paths carry the same amount of gradients.
- We sample individual paths of a certain length and measure the norm of gradients that arrives at the input.
- The gradient magnitude of a path decreases exponentially with the number of modules.



# ResNets – exponential ensembles of relatively shallow networks

- We multiply the frequency of each path length with its expected gradient magnitude.
- Almost all of the gradient updates come from paths relatively shallow.



# Discussion

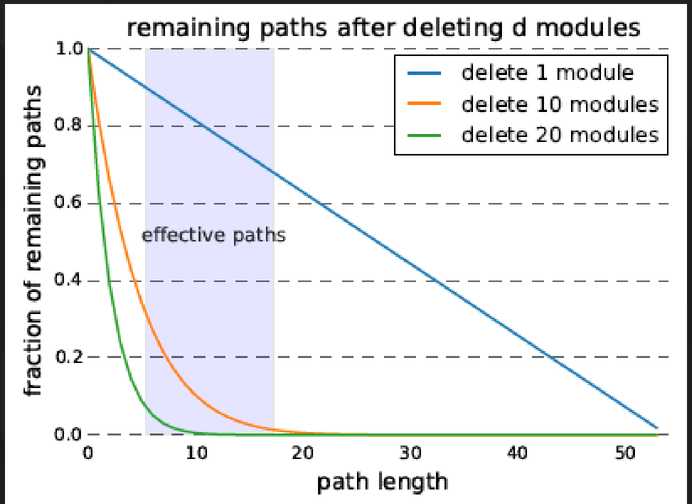
- Removing residual modules mostly removes long paths
- Connection to highway networks
- Effect of stochastic depth training procedure

# Removing residual modules mostly removes long paths

- Deleting d residual modules from a network of n, the fraction of paths remaining per path length x is given by

$$\text{fraction of remaining paths of length } x = \frac{\binom{n-d}{x}}{\binom{n}{x}}$$

- The deletion of residual modules mostly affects the long paths.



# Connection to highway networks

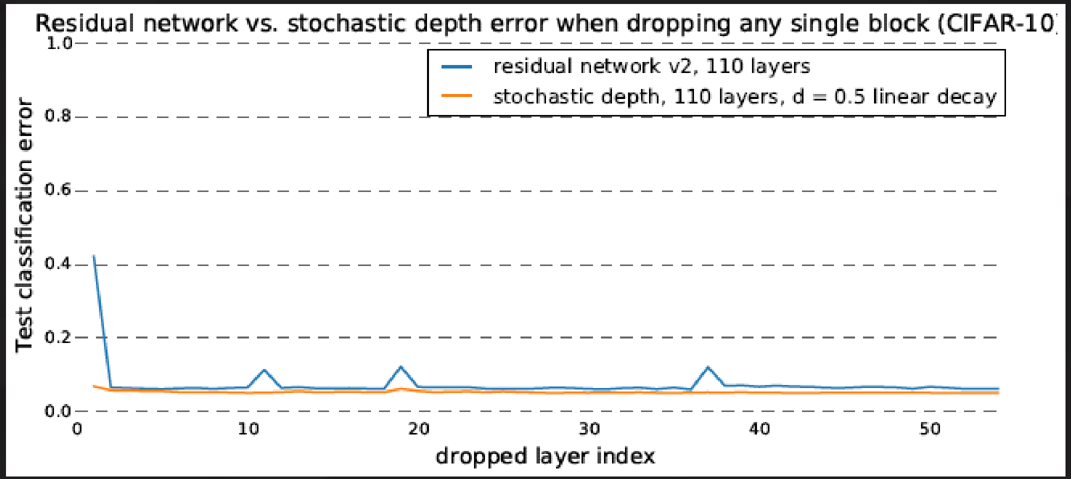
- The exponential nature of ResNet arises when data can flow both paths within a residual block at once.
- For highway networks, it’s not the case.

$$y_{i+1} \equiv f_{i+1}(y_i) \cdot t_{i+1}(y_i) + y_i \cdot (1 - t_{i+1}(y_i))$$

- In highway networks, gates commonly deviate from $t_i()=0.5$, reducing the number of expected paths.
- Highway networks are biased to send data through the skip connection, meaning they use short paths at the cost of decreasing expected multiplicity.

# Effect of stochastic depth training procedure

- In stochastic depth training, a random subset of the residual modules is selected for each mini-batch during training both forward and backward.
- The training method does not affect the multiplicity of the network because all the paths are available during the training.
- It shortens the paths seen during training and encourage the paths to independently produce good results.



# Conclusion

- It is not depth, but the ensemble that makes residual networks strong.
- ResNet pushes the limit of network multiplicity rather than depth.
- The paths that contribute gradient are very short compared to the overall depth of the network.
- First step, further exploration.

# Other literatures in ResNet

- Inception ResNet

  **“Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning”, arxiv 2016/8/23, Christian Szegedy et al.**

- ResNet in ResNet

  **“ResNet in ResNet: Generalizing Residual Architectures”, arxiv 2016/3/25, Sasha Targ et al.**

- Width v.s. Depth

  **“Wide Residual Networks”, arxiv 2016/5/23, Sergey Zagoruyko et al.**