AI and Deep Learning

4-5 ResNet

Zhonglei Wang

WISE, SOE, CHOW

XMU

(Version v1)

Contents

Motivation

1. Won the 1st place of ImageNet ILSVRC 2015 classification competition
2. Theoretically, the training error should go down as the number of layers increases
3. However, it is not the case

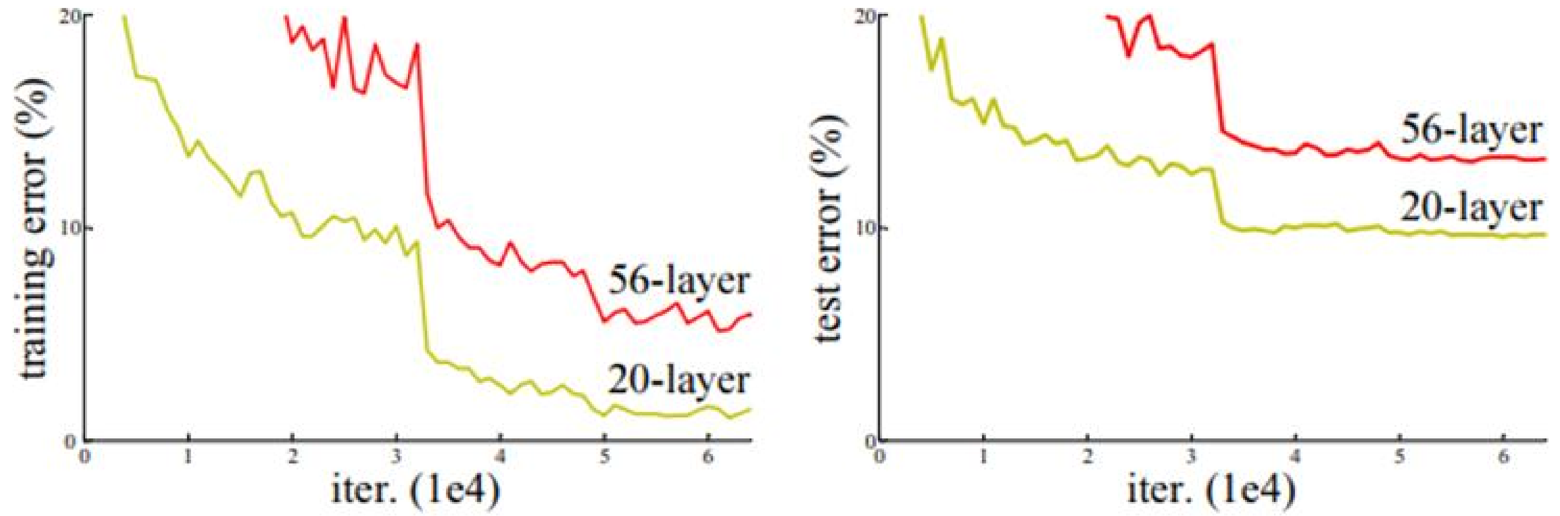


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

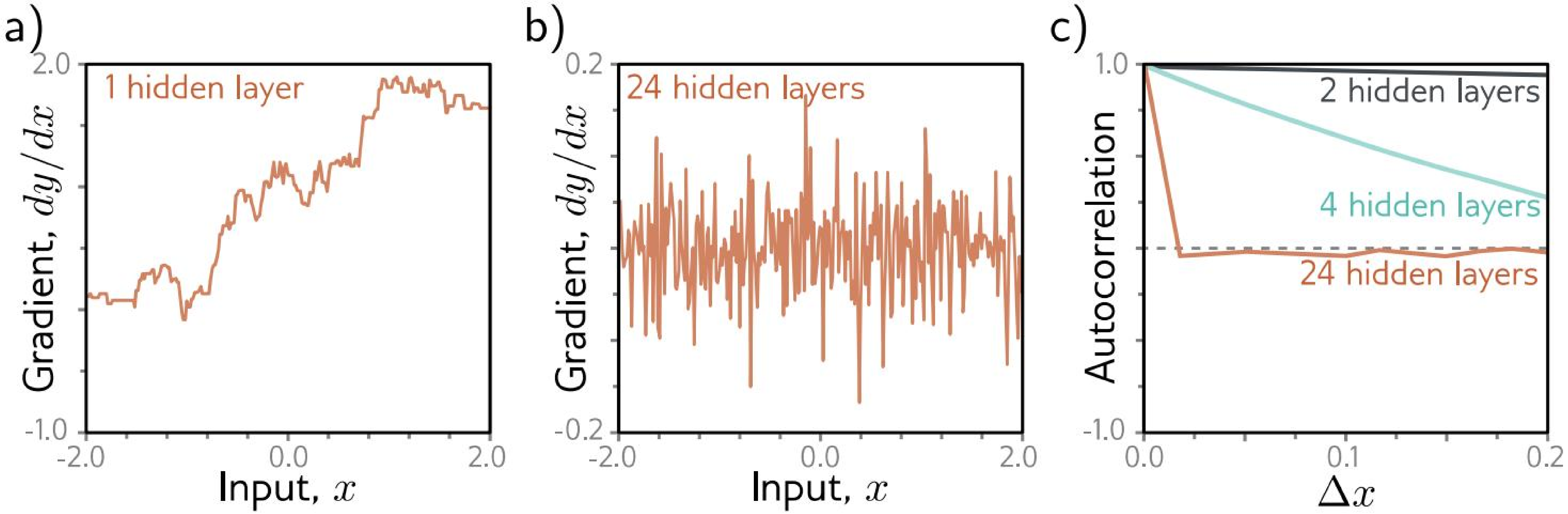
[K. He, X. Zhang, S. Ren and J. Sun, Deep Residual Learning for Image Recognition, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778]

Motivation

1. Gnerally, more complex models may lead to overfitting, with worse performance in test data
2. However, it may decrease the training error due to “overfitting”
3. That means, the problem is training deeper neural networks, not their generalizatoin
4. This phenomenon is not completely understood, but one conjecture is at initialization
   - Loss gradients change unpredictably when we modify parameters in early network layers
   - With reasonable initialization, we can avoid gradient exploding or vanishing
   - However, the derivative assumes infinitesimal step size
     - In practice, our step size is finite
     - It is problematic if the loss surface looks like enormous range of tiny mountains rather than a smooth one
   - It is supported by empirical observations of gradients in networks with a single input and output

Motivation

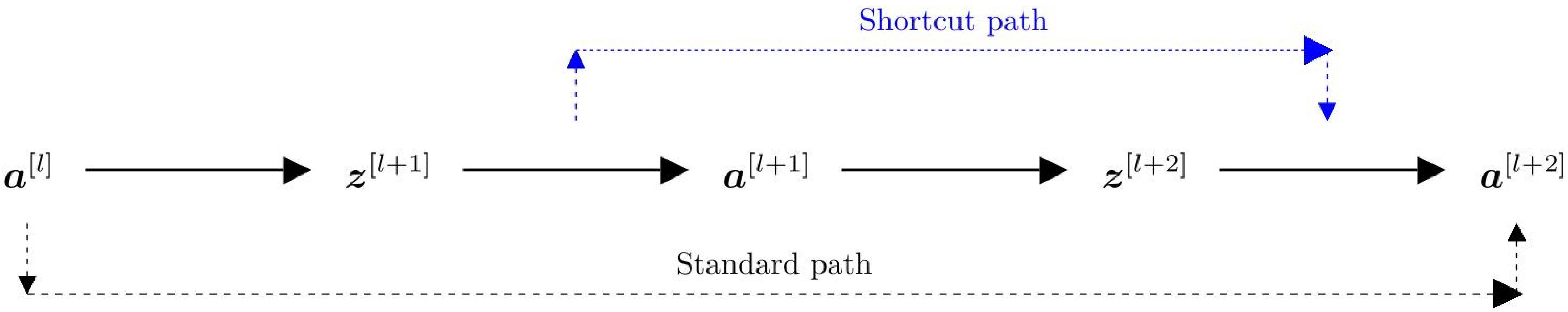
1. The following is Figure 11.3 of Prince (2024)



Motivation

1. We conclude
   - For shallower networks, the gradient of output with respect to the input changes slowly
   - For deeper networks, however, it is not this case
2. This is captured by the autocorrelation function of the gradient
   - For shallower networks, nearby gradients are highly autocorrelated
   - For deeper networks, however, it is not this case
3. This is termed the shattered gradients phenomenon
4. ResNet solves this problem by changing the autocorrelation of gradients

Structure



$$\boldsymbol{z}^{[l+1]} = \boldsymbol{W}^{l+1} \cdot \boldsymbol{a}^{[l]} + \boldsymbol{b}^{[l+1]} \qquad \boldsymbol{a}^{[l+1]} = \sigma^{[l+1]}(\boldsymbol{z}^{[l+1]}) \qquad \boldsymbol{z}^{[l+2]} = \boldsymbol{W}^{l+2} \cdot \boldsymbol{a}^{[l+1]} + \boldsymbol{b}^{[l+2]} \qquad \boldsymbol{a}^{[l+2]} = \sigma^{[l+2]}(\boldsymbol{z}^{[l+2]})$$

1. Shortcut path

$$\boldsymbol{a}^{[l+2]} = \sigma^{[l+2]}(\boldsymbol{z}^{[l+2]} + \boldsymbol{z}^{[l+1]})$$

2. Thus, $\boldsymbol{z}^{[l+2]}$ models the “residual” associated with $\boldsymbol{z}^{[l+1]}$ and $\boldsymbol{z}^{[l+2]}$

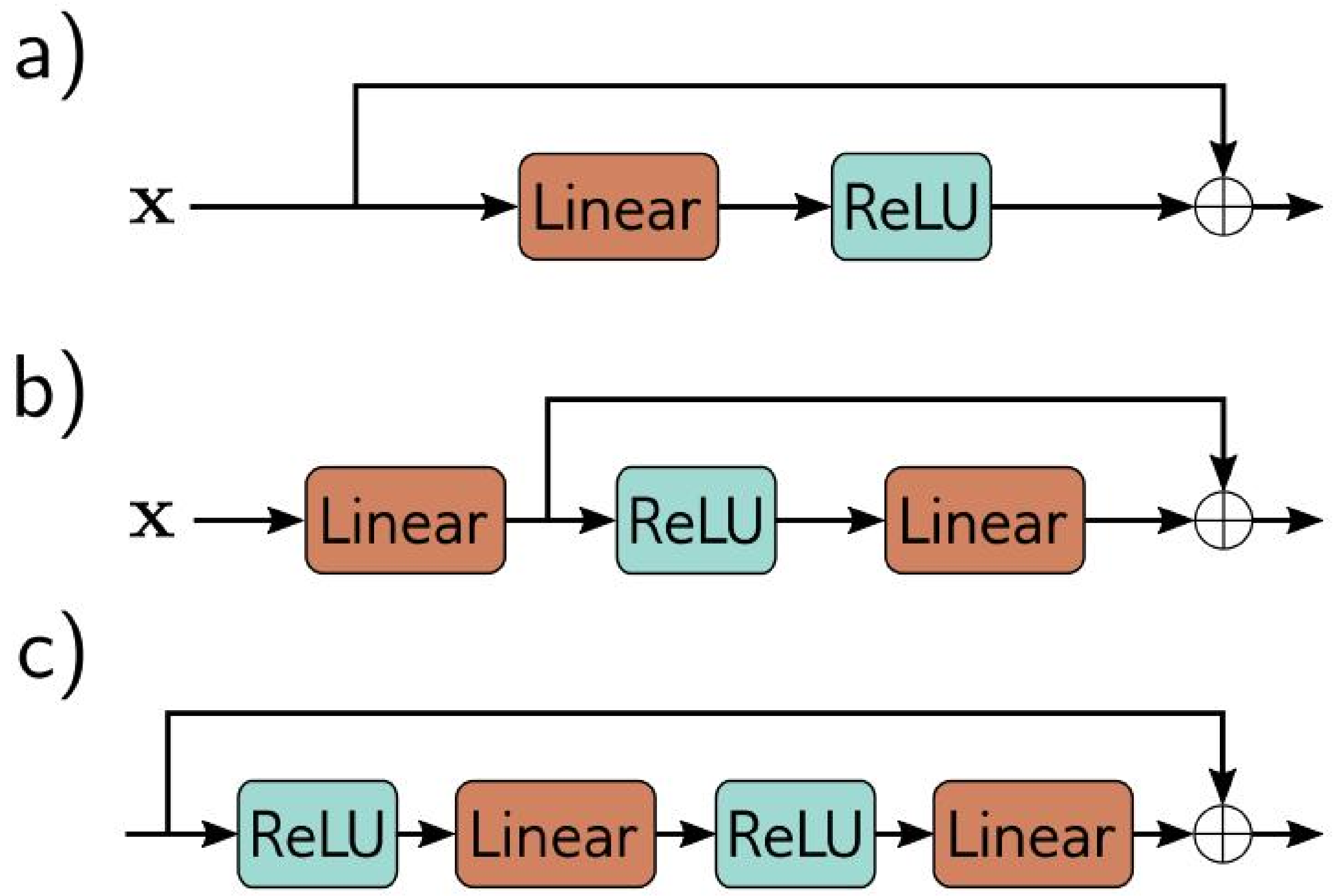
Remarks

1. More general, the length of $\boldsymbol{z}^{[l+2]}$ may be different from that of $\boldsymbol{z}^{[l+1]}$
2. This problem can be easily solved by introducing a new parameter$\boldsymbol{W}_s^{[l]}$

$$\boldsymbol{a}^{[l+2]} = \sigma^{[l+2]}(\boldsymbol{z}^{[l+2]} + \boldsymbol{W}_s^{[l]} \cdot \boldsymbol{z}^{[l+1]})$$

Remarks

1. The following if Figure 11.5 of Prince (2024)



Remarks

1. Notice that the output after ReLU activation is usually nonnegative
2. Thus, if we the shortcut path join right after ReLU activation, we can only increase the values of "input"
3. Residual connection usually joins two linear transformation results

Remarks

1. Residual connection can be used to deepen neural networks
2. Thus, we may suffer from gradient vanishing or exploding (exponentially)
3. A typical technique to alleviate this difficulty is to use batch normalization

Summary

1. ResNet reformulates the learning task as “preserving the original information + learning incremental corrections” by a shortcut path
2. Pay attention to the start and end points of the shortcut path, if ReLU is the activation function
3. For deep learning models, ResNet cannot solve all problems about gradient