THE DIFFERENCE BETWEEN CNN AND DL

Yuxuan Luo, Zhongwu Xie

Outlook

- Introduction of DL
- The difference of CNN between MLP
- Other Deep Learning models

Deep Learning

No formal definition.

Models contain several features may be the deep learning model:

- contains a collection of statistical machine learning techniques
- used to learn feature hierarchies
- often based on artificial neural networks

Generally, when the model more than 5 layers that is Deep learning model. There are many deep learning models .

e.g.

Multi Layers Perception, Convolutional Neural Network, Residual Network,

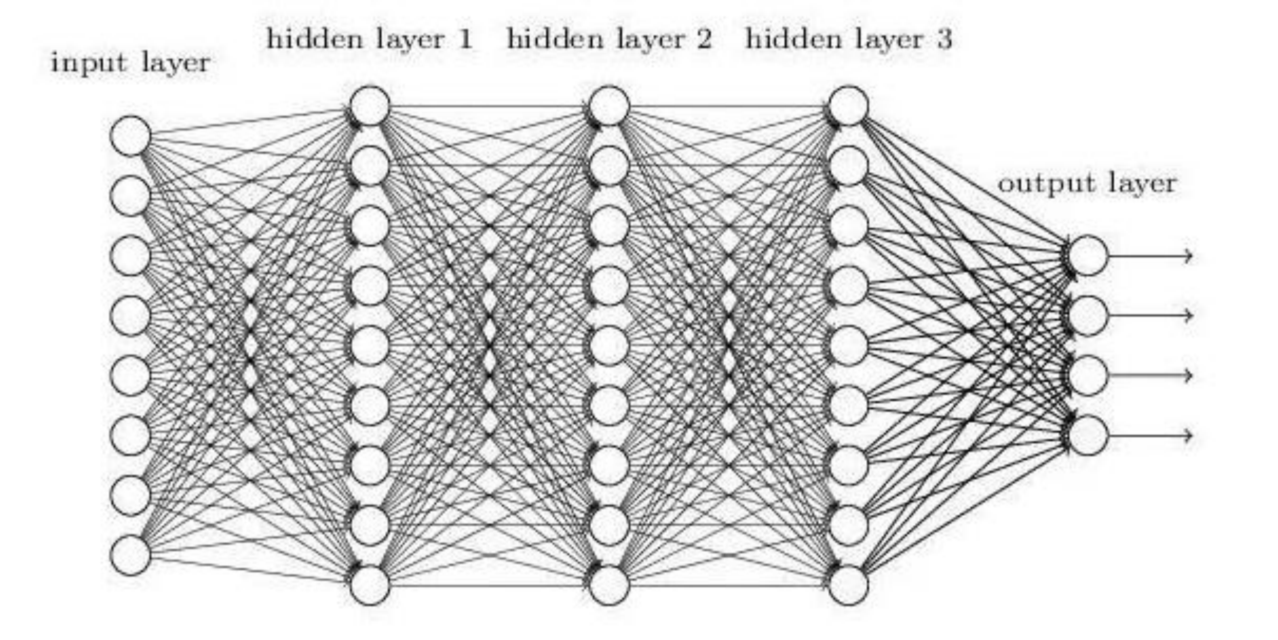
Deep Belief Network, Recursive Neural Network and etc.

CNN is one of famous deep learning model.

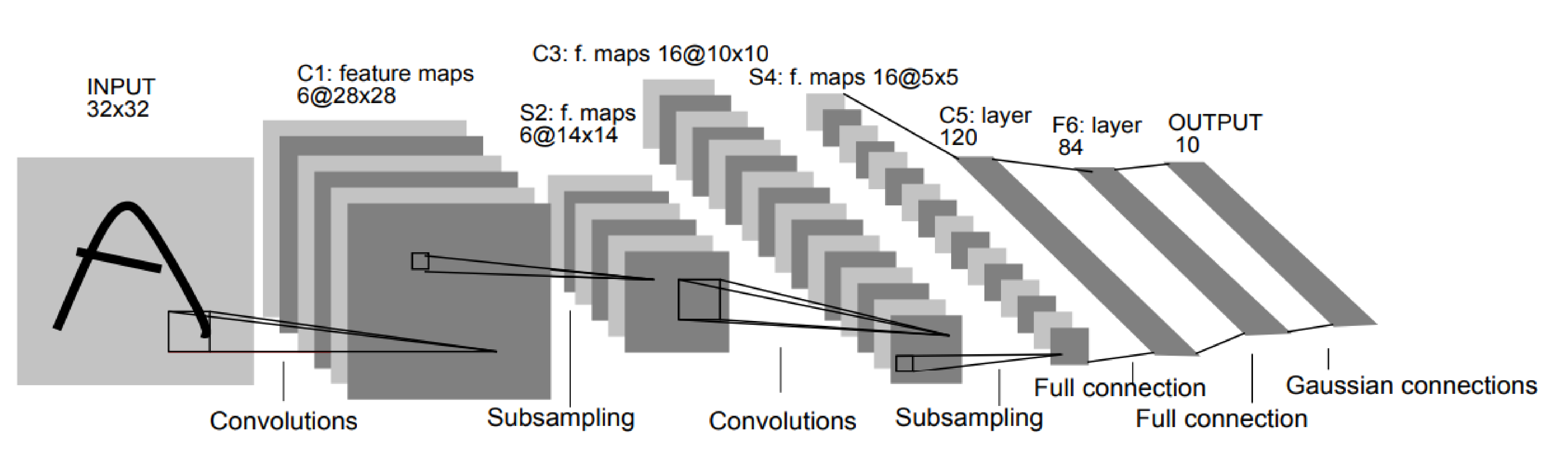
Differences:

1 datasets
2 features extracting
3 parameters-sharing
4 sparsity of connections

MLP



CNN



Input Volume (+pad 1) (7x7x3)

x[:,:,0]

0	0	0	0	0	0	0
0	0	1	1	0	2	0
0	2	2	2	2	1	0
0	1	0	0	2	0	0
0	0	1	1	0	0	0
0	1	2	0	0	2	0
0	0	0	0	0	0	0

x[:,:,1]

0	0	0	0	0	0	0
0	1	0	2	2	0	0
0	0	0	0	2	0	0
0	1	2	1	2	1	0
0	1	0	0	0	0	0
0	1	2	1	1	1	0
0	0	0	0	0	0	0

x[:,:,2]

0	0	0	0	0	0	0
0	2	1	2	0	0	0
0	1	0	0	1	0	0
0	0	2	1	0	1	0
0	0	1	2	2	2	0
0	2	1	0	0	1	0
0	0	0	0	0	0	0

Filter W0 (3x3x3)

w0[:,:,0]

-1	1	0
0	1	0
0	1	1

w0[:,:,1]

-1	-1	0
0	0	0
0	-1	0

w0[:,:,2]

0	0	-1
0	1	0
1	-1	-1

Bias b0 (1x1x1)

b0[:,:,0]

1

Filter W1 (3x3x3)

w1[:,:,0]

1	1	-1
-1	-1	1
0	-1	1

w1[:,:,1]

0	1	0
-1	0	-1
-1	1	0

w1[:,:,2]

-1	0	0
-1	0	1
-1	0	0

Bias b1 (1x1x1)

b1[:,:,0]

0

Output Volume (3x3x2)

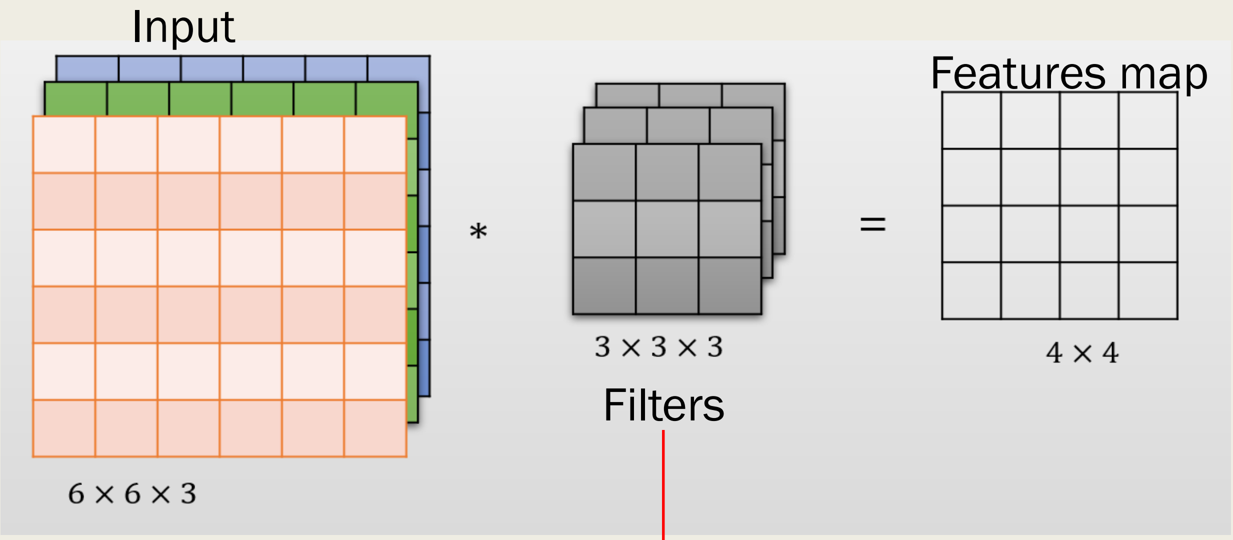
o[:,:,0]

6	7	5
3	-1	-1
2	-1	4

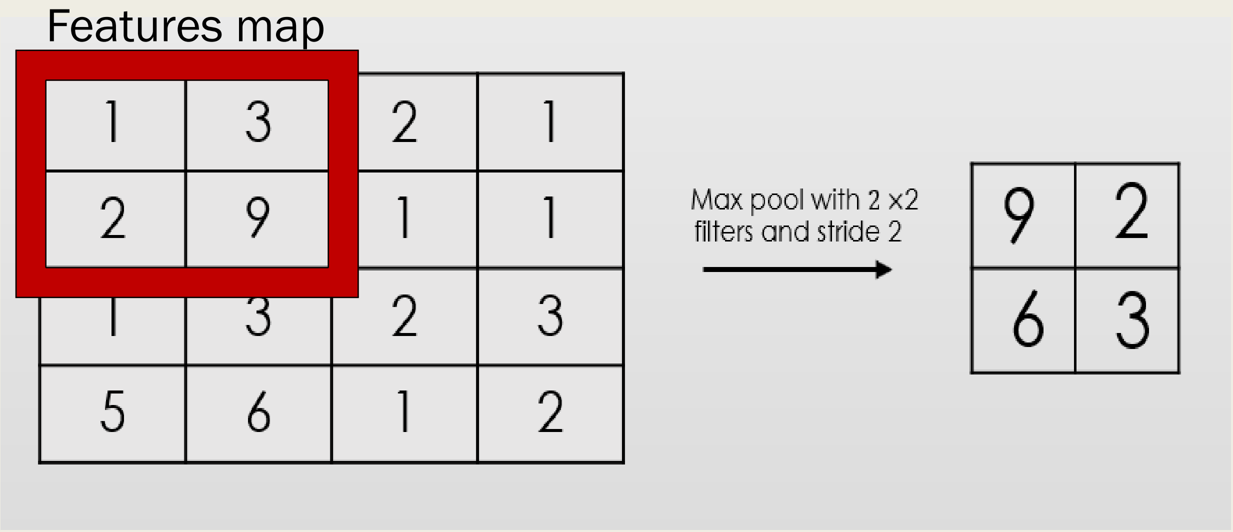
o[:,:,1]

2	-5	-8
1	-4	-4
0	-5	-5

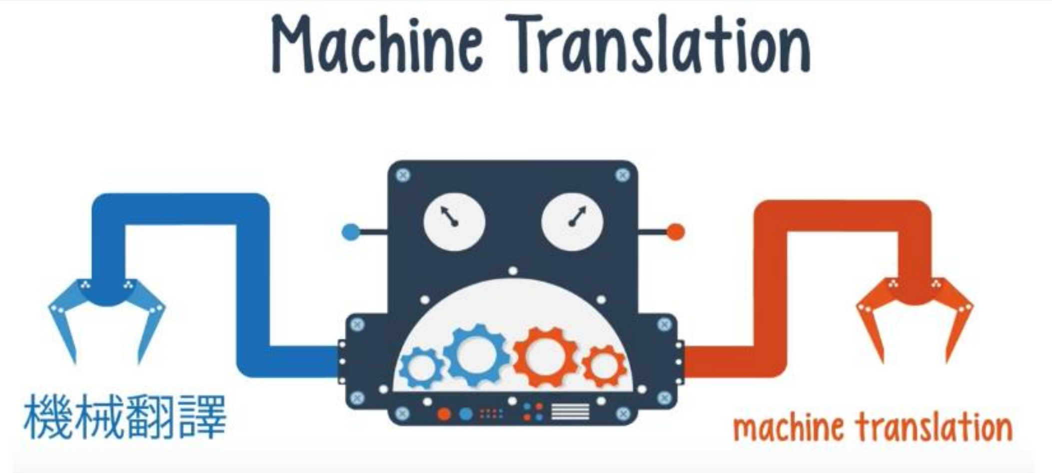
toggle movement

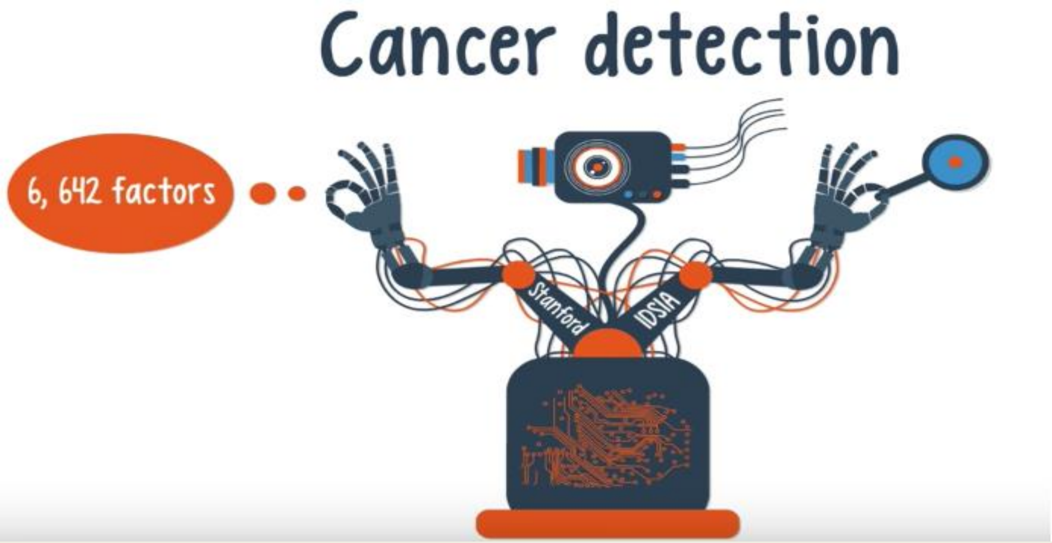


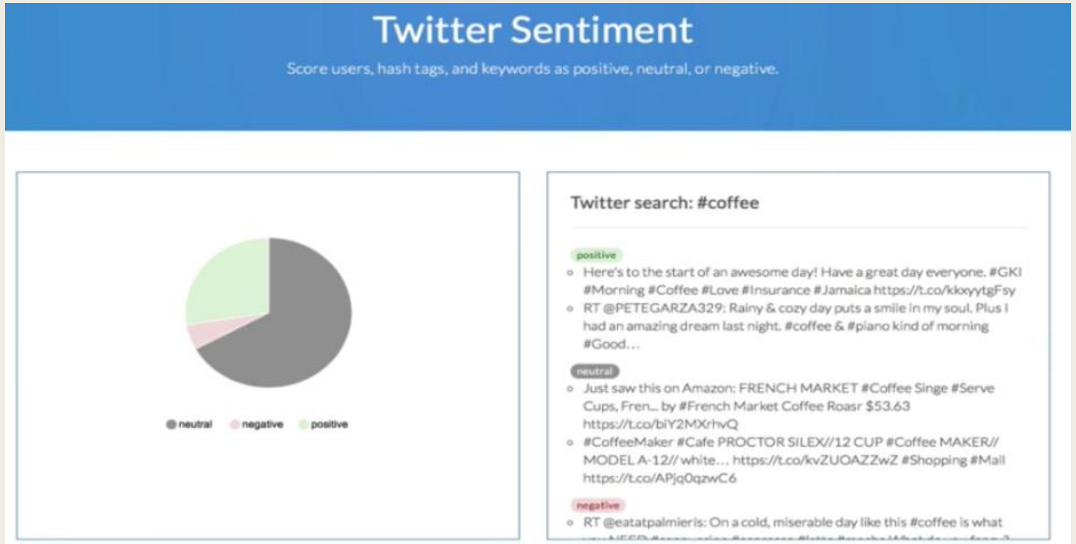
CNN update the Filter weight so that
it can extract features correctly, but it share the
weight in extracting the same kind of features.



Application Area



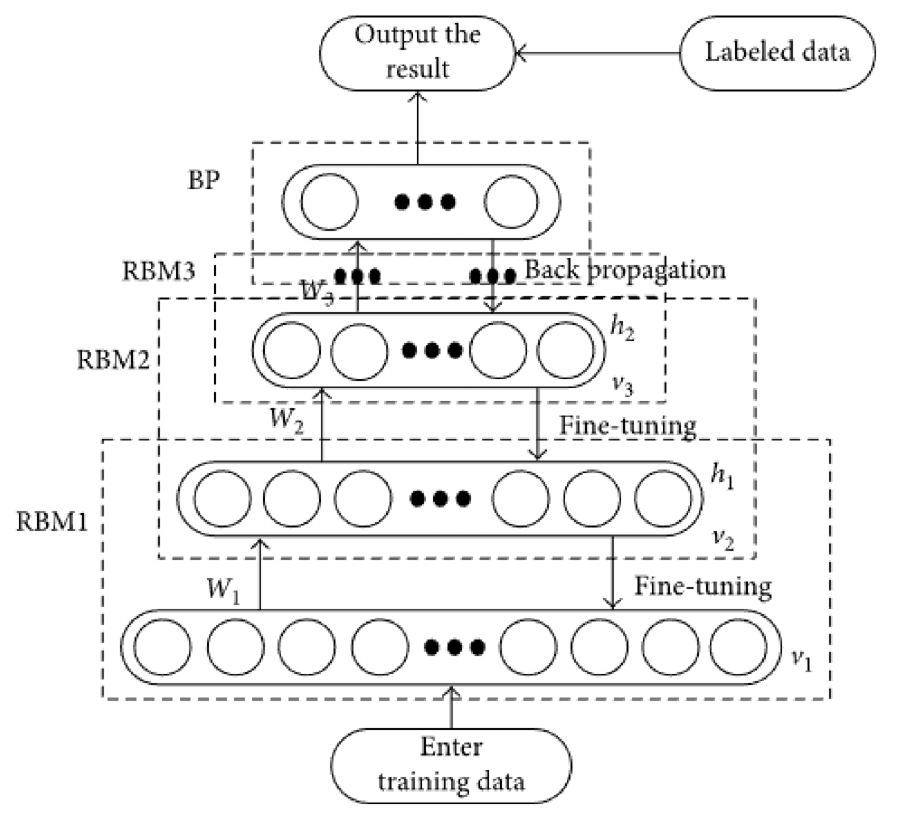




Deep Learning Model

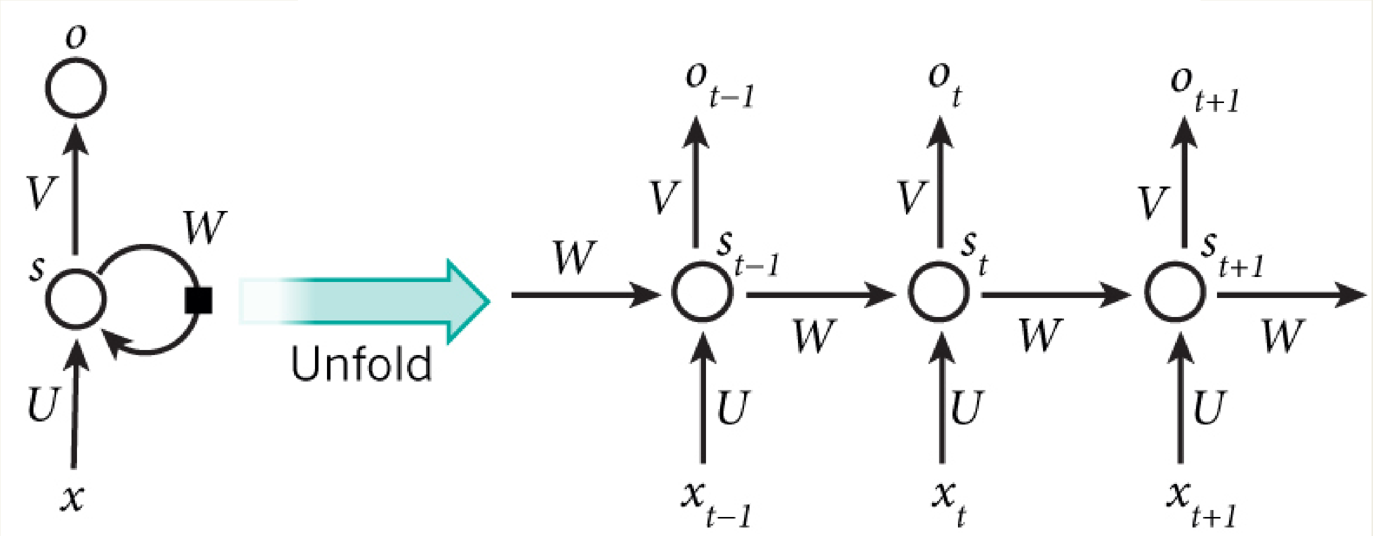
1.Deep Belief Network

Is a Generative model,
consist of several Restricted Boltzmann Machines.
Unsupervised learning, pre-learning, fine-tune to train models.



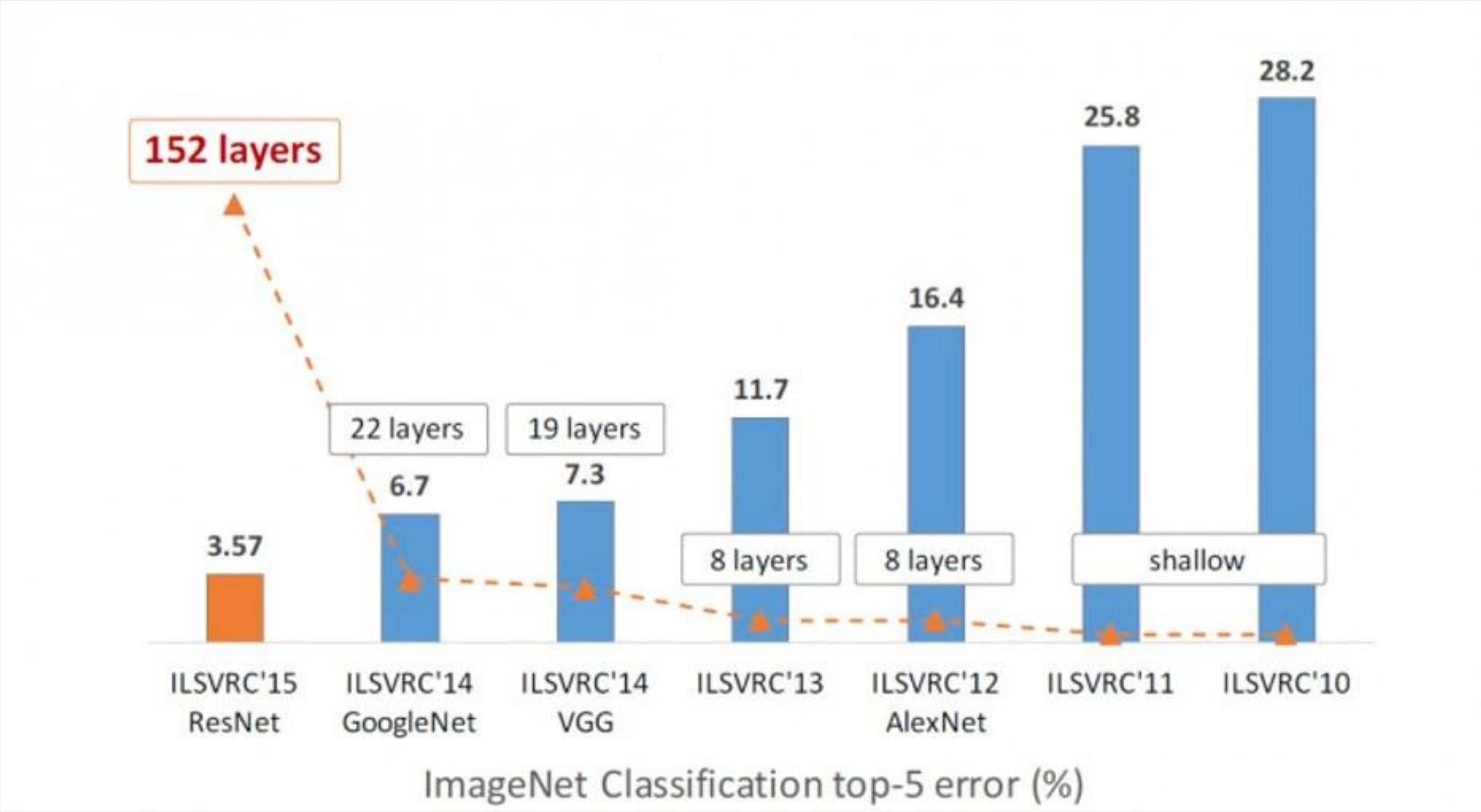
2.Recursive Neural Network

Using in language modeling ,
generating text, Machine Translation.
Make use of sequential information
and dividing into a tree



3.Residual Neural Networks

Revolution of Depth



What is the advantages of deep model with more layers?

- The “level” of feathers will enrich, when the depth of neural network increase.
- With more deeper layers, the network has more powerful representational ability.

Driven by the significance of depth, a question arises :

- The problem of vanishing/exploding gradients.
- Degradation problem.

Vanishing/exploding gradients

---the chain rule

$$x,\ w^{[1]},\ b^{[1]} \to \boxed{z^{[1]} = w^{[1]}x + b^{[1]}} \to \boxed{a^{[1]} = g(z^{[1]})} \to \boxed{z^{[2]} = w^{[2]}a^{[1]} + b^{[2]}} \to \boxed{a^{[2]} = g(z^{[2]})} \to \boxed{z^{[3]} = w^{[3]}a^{[2]} + b^{[3]}} \to \boxed{a^{[3]} = g(z^{[3]})} \to \boxed{L(a^{[2]}, y)}$$

$$w^{[2]},\ b^{[2]} \to \boxed{z^{[2]} = w^{[2]}a^{[1]} + b^{[2]}} \qquad w^{[3]},\ b^{[3]} \to \boxed{z^{[3]} = w^{[3]}a^{[2]} + b^{[3]}}$$

$$dw^{[3]} = \frac{\partial L(a,y)}{\partial w^{[3]}} = \frac{\partial L(a,y)}{\partial a^{[3]}} \times \frac{\partial a^{[3]}}{\partial z^{[3]}} \times \frac{\partial z^{[3]}}{\partial w^{[2]}} = \frac{\partial L(a,y)}{\partial a^{[3]}} \times g'(z^{[3]}) \times a^{[2]}$$

$$dw^{[2]} = \frac{\partial L(a,y)}{\partial w^{[2]}} = \frac{\partial L(a,y)}{\partial a^{[3]}} \times \frac{\partial a^{[3]}}{\partial z^{[3]}} \times \frac{\partial z^{[3]}}{\partial a^{[2]}} \times \frac{\partial a^{[2]}}{\partial z^{[2]}} \times \frac{\partial z^{[2]}}{\partial w^{[2]}} = \frac{\partial L(a,y)}{\partial a^{[3]}} \times g'(z^{[3]}) \times w^{[3]} \times g'(z^{[2]}) \times a^{[1]}$$

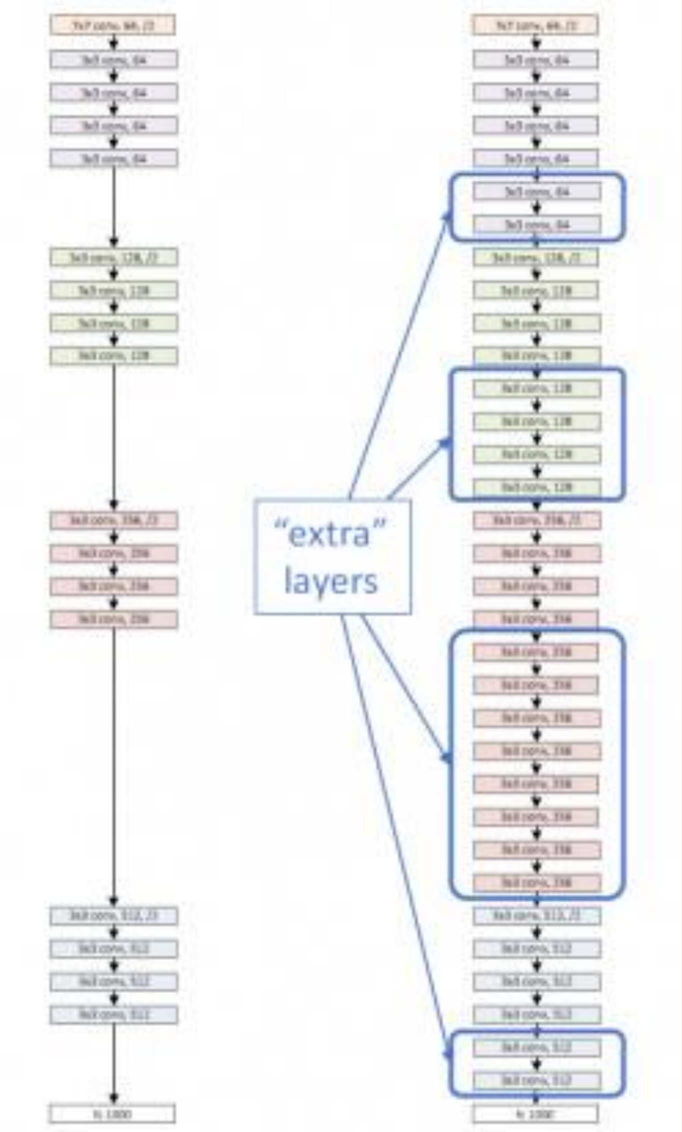
$$dw^{[1]} = \frac{\partial L(a,y)}{\partial w^{[2]}} = \frac{\partial L(a,y)}{\partial a^{[3]}} \times \frac{\partial a^{[3]}}{\partial z^{[3]}} \times \frac{\partial z^{[3]}}{\partial a^{[2]}} \times \frac{\partial a^{[2]}}{\partial z^{[2]}} \times \frac{\partial z^{[2]}}{\partial a^{[1]}} \times \frac{\partial a^{[1]}}{\partial z^{[1]}} \times \frac{\partial z^{[1]}}{\partial w^{[1]}} = \frac{\partial L(a,y)}{\partial a^{[3]}} \times g'(z^{[3]}) \times w^{[3]} \times g'(z^{[2]}) \times w^{[2]} \times g'(z^{[1]}) \times x$$

If the value of weights are very small, the gradients will vanish. If the value is greater than 1,the gradients will be very large.

Degradation problem

- A solution by construction:
  - original layers : copied from a learned shallower model
  - Extra layers :learn to set as identity
  - At least the same training error
- Richer solution space
- A deeper model should not have higher training error

But the result is …



Degradation problem

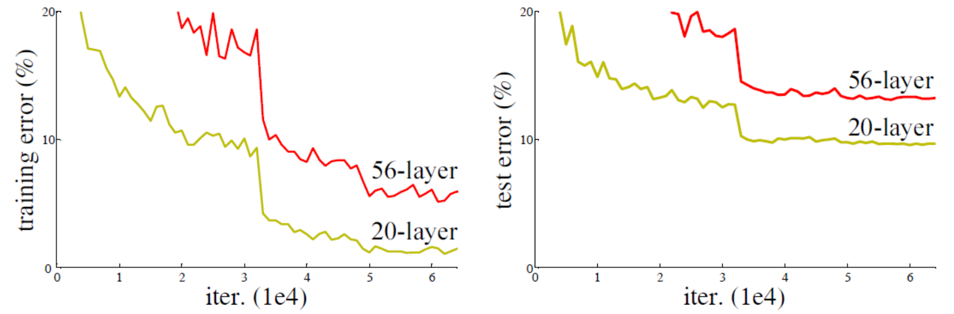
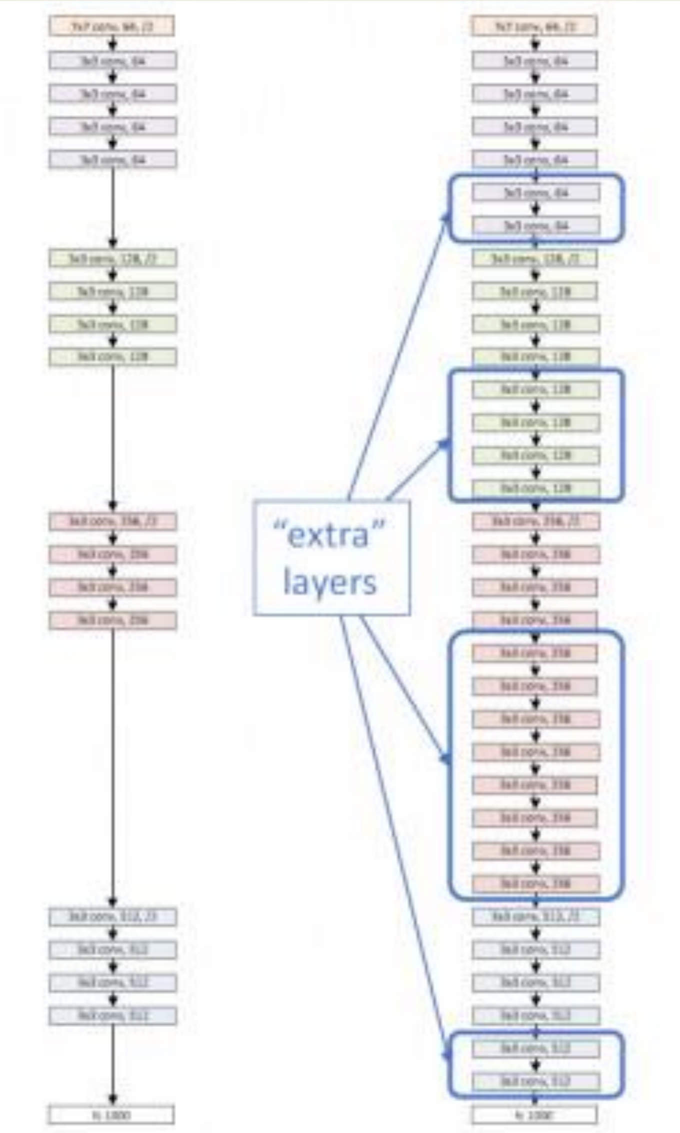


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

- "overly deep" plain nets have higher training error
- A general phenomenon, observed in many datasets

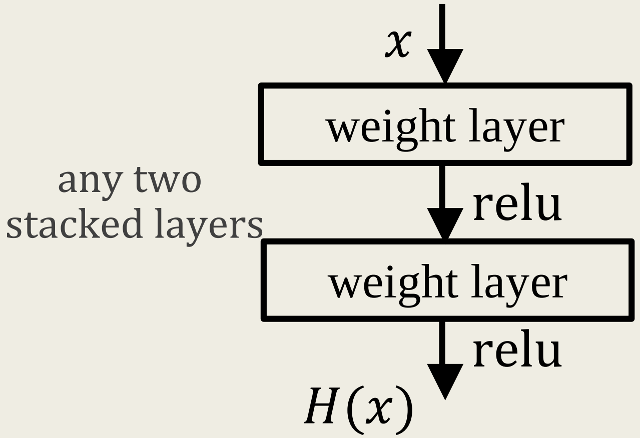
But the problem doesn't cause by overfitting.

Optimization difficulties : solvers cannot find the solution when going deeper --- the solvers might have difficulties in approximating identity mappings by multiple nonlinear layers.



A building block

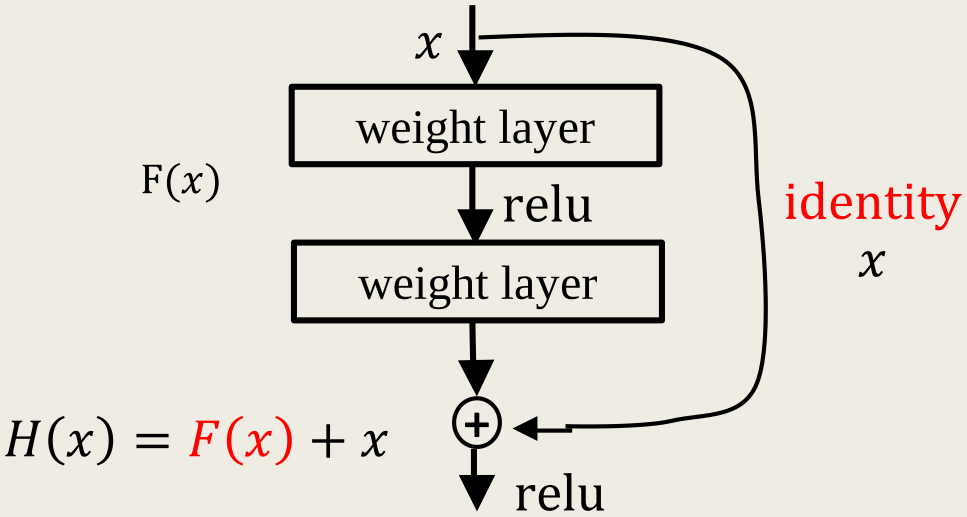
- Plaint net



$H(x)$ is any desired mapping,

hope the 2 weight layers fit $H(x)$

- Residual net



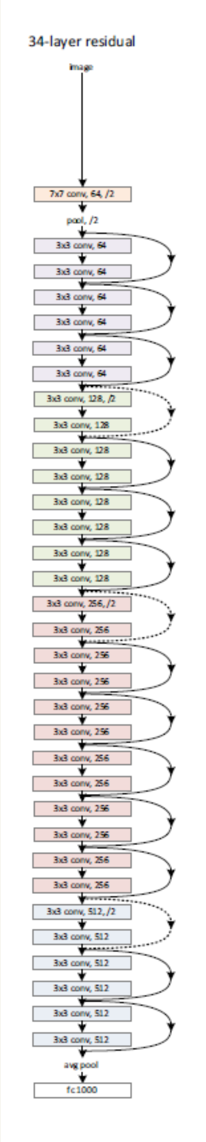
$H(x)$ is any desired mapping,

~~hope the 2 weight layers fit $H(x)$~~

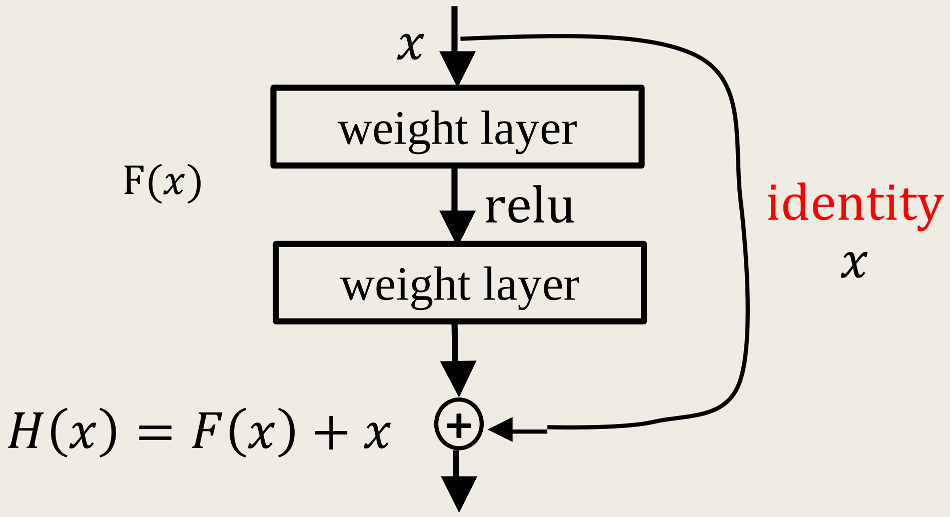
hope the 2 weight layers fit *F(x)*

Let H(x)=F(x)+x

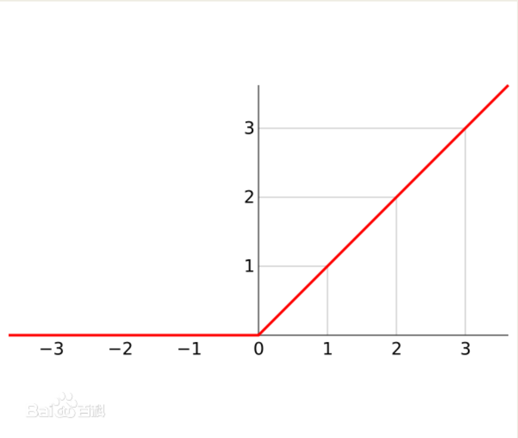
What the residual network looks like

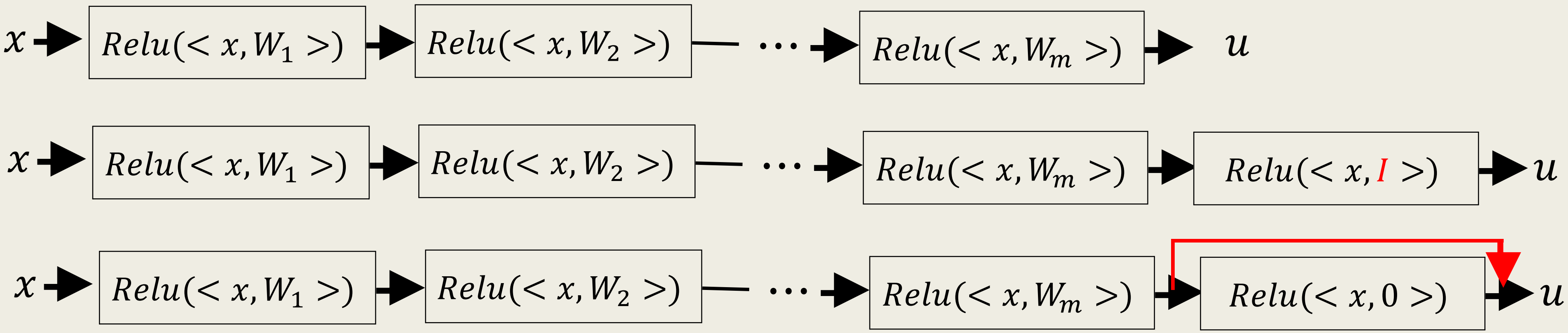


Why can the residual block learn identity mapping easier?



- It is more easier for the weights of two stacked layers to fit to zero metric than identity matrix.
- The initialization of weights.



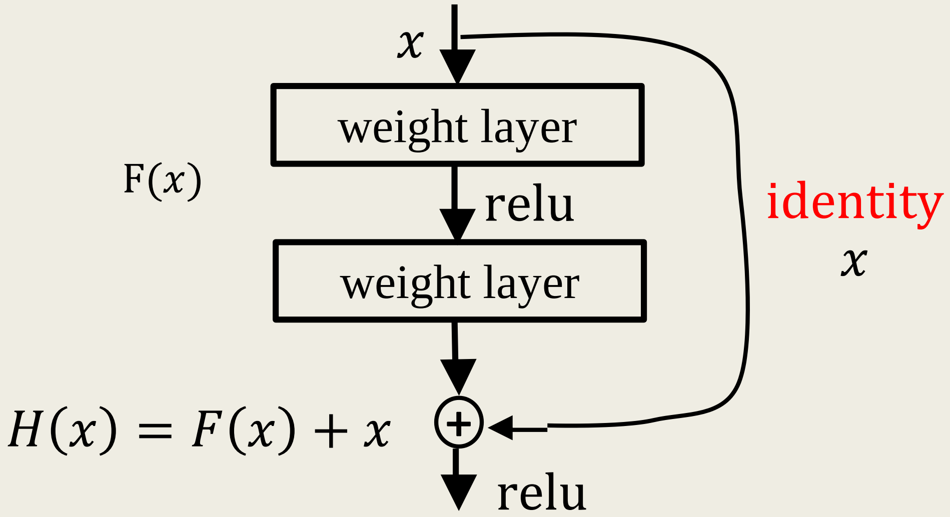


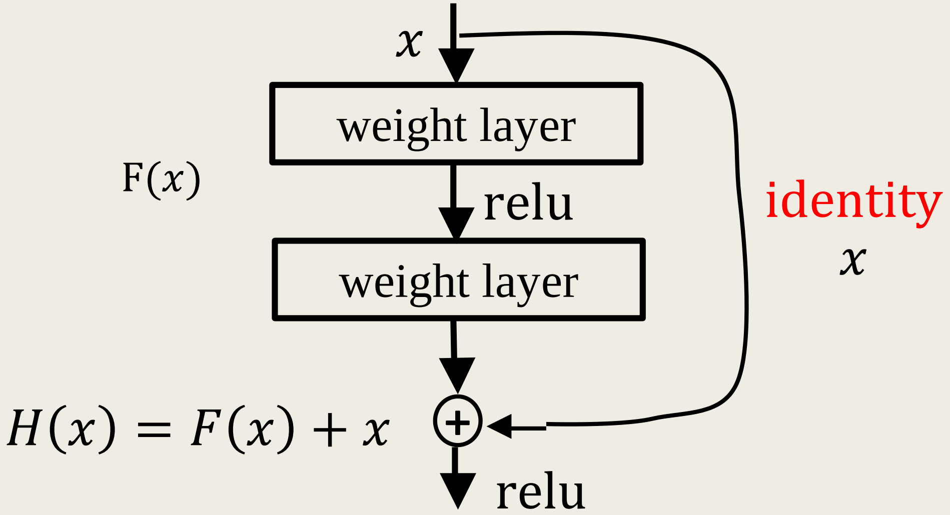
Whether have we addressed the two problems?

- The problem of vanishing/exploding gradients.
- Degradation problem.

Solve the problem of vanishing/exploding gradients.

- Residual net





- If identity were optimal , easy to set weights as 0.
- If optimal mapping is closer to identity,

  easier to find small fluctuations

$$y_l = h(x_l) + F(x_l, W_l)$$

$$x_{l+1} = f(y_l)$$

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

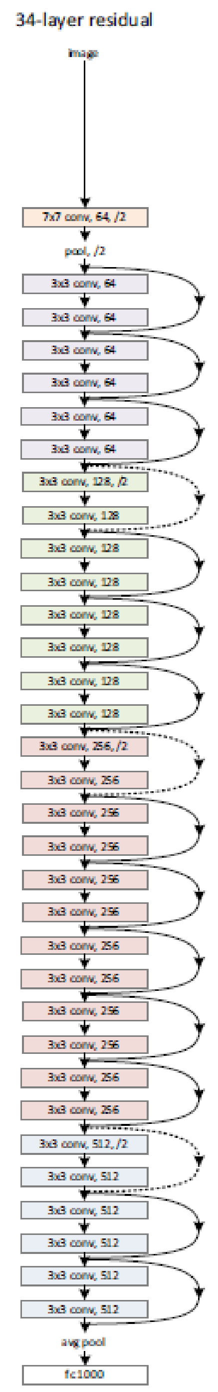
$$\mathbf{x}_{l+1} = \mathbf{x}_l + \mathcal{F}(\mathbf{x}_l, \mathcal{W}_l). \tag{3}$$

Recursively $(\mathbf{x}_{l+2} = \mathbf{x}_{l+1} + \mathcal{F}(\mathbf{x}_{l+1}, \mathcal{W}_{l+1}) = \mathbf{x}_l + \mathcal{F}(\mathbf{x}_l, \mathcal{W}_l) + \mathcal{F}(\mathbf{x}_{l+1}, \mathcal{W}_{l+1})$, etc.) we will have:

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

Eqn.(4) also leads to nice backward propagation properties. Denoting the loss function as $\mathcal{E}$, from the chain rule of backpropagation [9] we have:

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



Solve the problem of degradation to some extent.

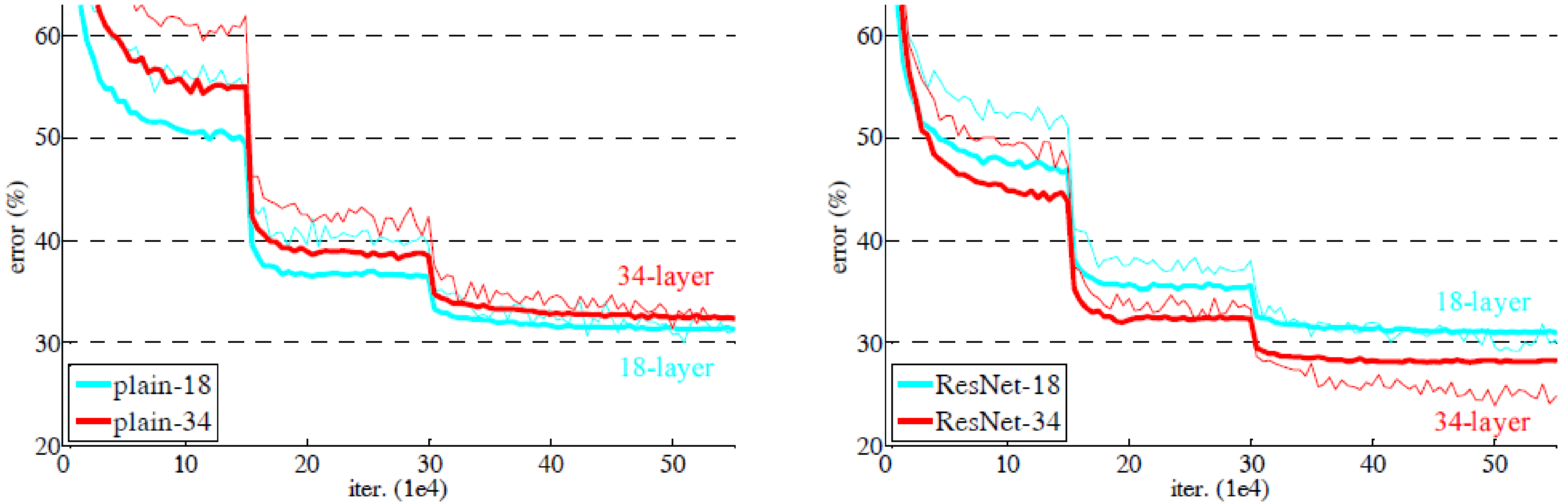
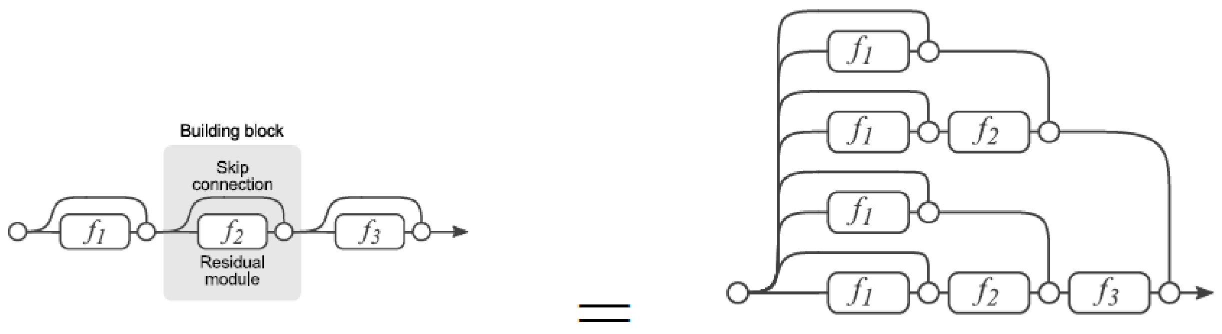


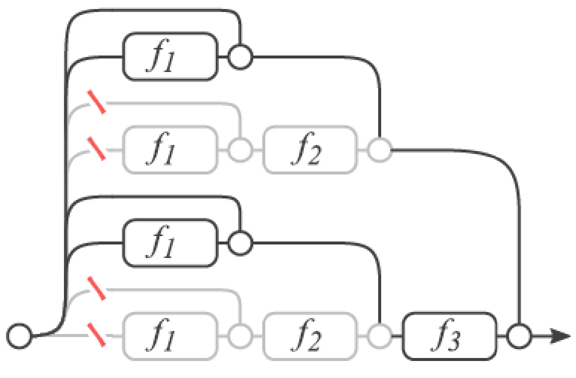
Figure 4. Training on **ImageNet**. Thin curves denote training error, and bold curves denote validation error of the center crops. Left: plain networks of 18 and 34 layers. Right: ResNets of 18 and 34 layers. In this plot, the residual networks have no extra parameter compared to their plain counterparts.

The intuition of Residual network.



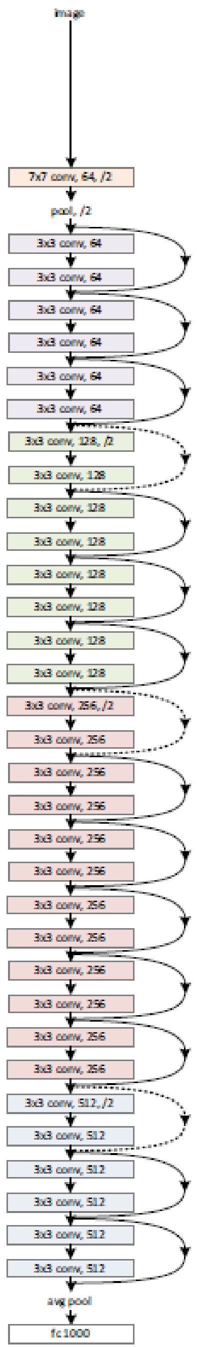
(a) Conventional 3-block residual network (b) Unraveled view of (a)

Residual networks can be viewed as a collection of many paths(it behaves like Ensembles of Relatively Shallow Networks).It consists of most moderate networks and a small portion of shallow and deep networks.

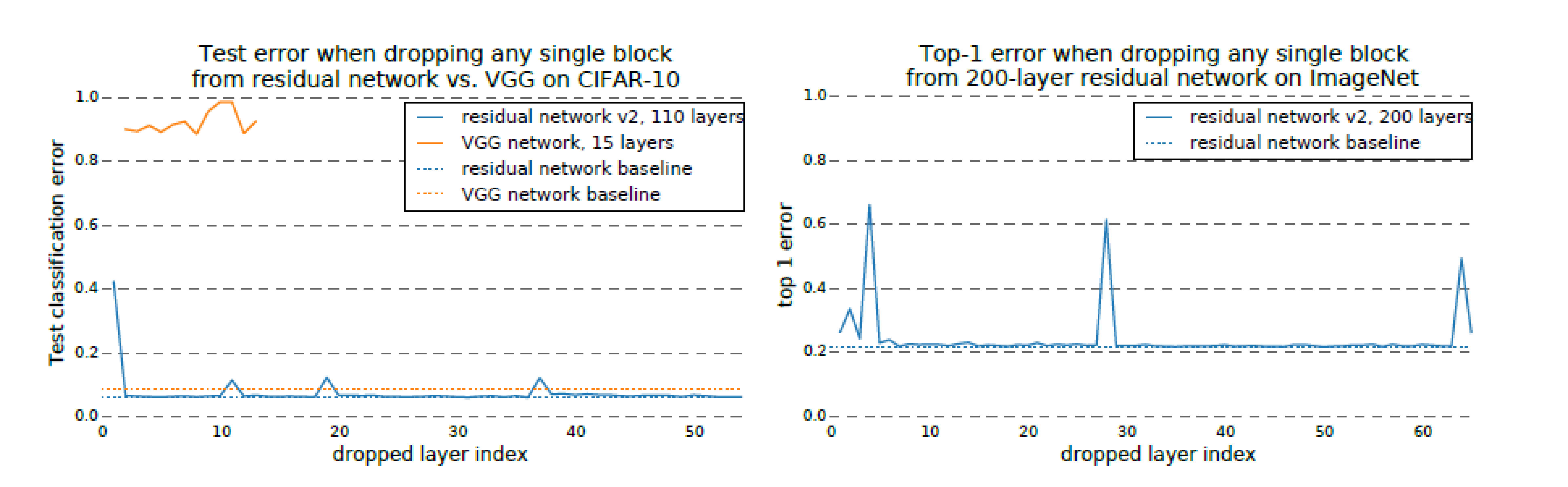


(a) Deleting f_2 from unraveled view

34-layer residual



The intuition of Residual network.



From the result of experiment :
The Residual Network looks seemingly very deep , but the network that actually works is not so deep.

It provides a way of thinking about model compression.

Reference

- He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.
- He K, Zhang X, Ren S, et al. Identity mappings in deep residual networks[C]//European conference on computer vision. Springer, Cham, 2016: 630-645.
- Veit A, Wilber M J, Belongie S. Residual networks behave like ensembles of relatively shallow networks[C]//Advances in Neural Information Processing Systems. 2016: 550-558.

Thanks for your attention.