Deeplearning

-ch13. CNN Architectures-

Industrial Management Engineering

2017.12.08

Jae Seung Song

CNN Architectures Contents

1. LeNet -5
2. AlexNet
3. GoogLeNet
4. ResNet

1. LeNet - 5

- Created by Yann leCun (1998).

- Define the concept of Convolutional Neural Network for first time.

- Imporve the problem of MLP that didn’t consider topology variations of letters.
  (It takes lots of time.)

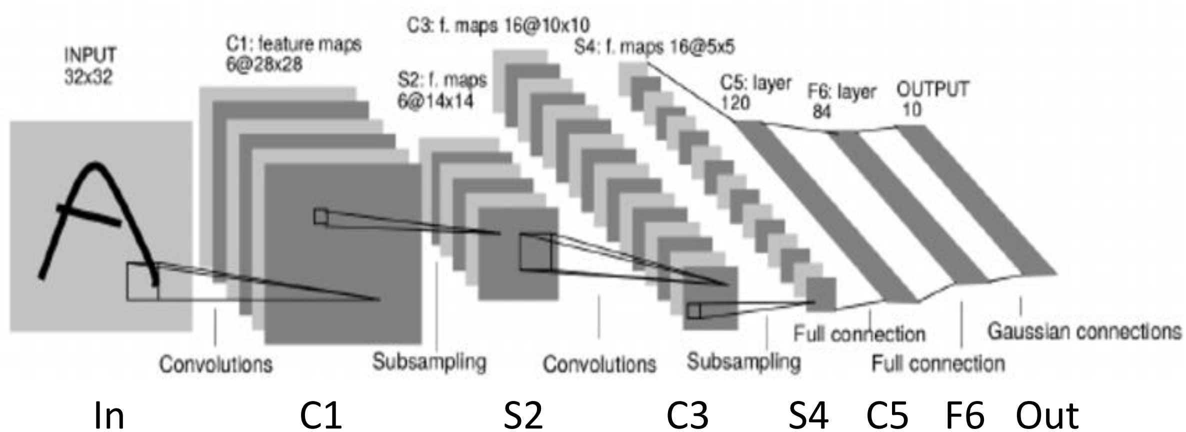
- The Purpose of MNIST classification

Table 13-1. LeNet-5 architecture

Layer	Type	Maps	Size	Kernel size	Stride	Activation
Out	Fully Connected	–	10	–	–	RBF
F6	Fully Connected	–	84	–	–	tanh
C5	Convolution	120	1 × 1	5 × 5	1	tanh
S4	Avg Pooling	16	5 × 5	2 × 2	2	tanh
C3	Convolution	16	10 × 10	5 × 5	1	tanh
S2	Avg Pooling	6	14 × 14	2 × 2	2	tanh
C1	Convolution	6	28 × 28	5 × 5	1	tanh
In	Input	1	32 × 32	–	–	–

Convolution layer : 3
Avg pooling : 2
Activation function : tanh or sigmoid function
Free parameter : 60,000

1. LeNet - 5



source : Gradient-Based Learning Applied to Document Recognition (IEEE 1998) – Yann LeCun

zero padding

- 28x28 ----> 32x32
- C3 : only connected 4 maps (instead of all 6 maps)
- Cost function : Euclidan distance (Recently, CrossEntropy function preferred)

2. AlexNet

- Won ImageNet ILSVRC challenge 2012 (Alex Khrizevsky)
- Initiate GPU technology in CNN (parallel processing)
- A structure similar to LeNet-5 but more deeper, larger

Table 13-2. AlexNet architecture

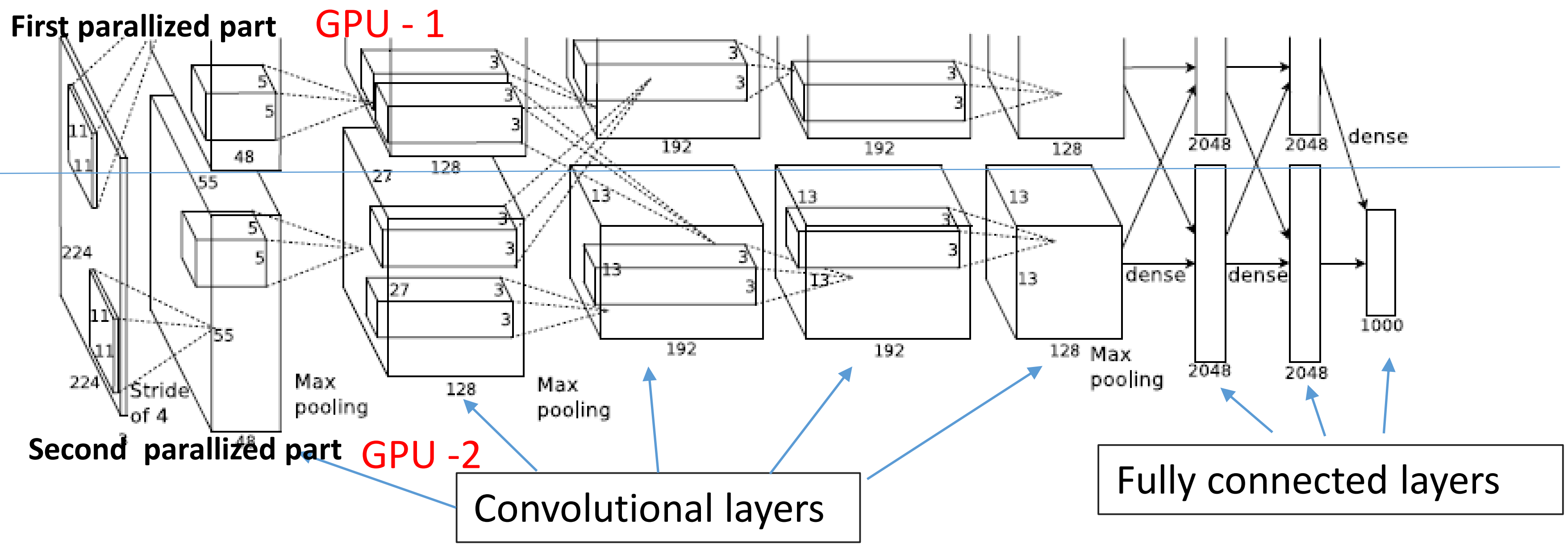
Layer	Type	Maps	Size	Kernel size	Stride	Padding	Activation
Out	Fully Connected	–	1,000	–	–	–	Softmax
F9	Fully Connected	–	4,096	–	–	–	ReLU
F8	Fully Connected	–	4,096	–	–	–	ReLU
C7	Convolution	256	13 × 13	3 × 3	1	SAME	ReLU
C6	Convolution	384	13 × 13	3 × 3	1	SAME	ReLU
C5	Convolution	384	13 × 13	3 × 3	1	SAME	ReLU
S4	Max Pooling	256	13 × 13	3 × 3	2	VALID	–
C3	Convolution	256	27 × 27	5 × 5	1	SAME	ReLU
S2	Max Pooling	96	27 × 27	3 × 3	2	VALID	–
C1	Convolution	96	55 × 55	11 × 11	4	SAME	ReLU
In	Input	3 (RGB)	224 × 224	–	–	–	–

Convolution layer : 5
Max pooling : 2
Activation function : ReLU
Freeparameter : 60,000,000

More complex than LeNet-5

2. AlexNet

Performance details



(1) Overlapped pooling
(2) Two regularization approaches
    a. Dropout
    b. Data augmentation
(3) Select ReLU activation function
    -> Local response normalization

source : ImageNet Classification with Deep Convolutional Neural Networks(2012) – Alex Krizhevsky

2. AlexNet

Performance details

(1) Overlapped pooling

Pooling is performed while overlapping. Use 3x3 kernel and 2 stride

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

Max pooling(2x2), 2strides

1.00	0.33	0.55	0.33
0.33	1.00	0.33	0.55
0.55	0.33	1.00	0.11
0.33	0.55	0.11	0.77

Max pooling , Avg pooling method

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

Max overlapped pooling(3x3), 2strides

1	**1**	**0.55**
1	**1**	**1**
0.55	**1**	**1**

Max Overapped method

Reduce overfitting problem!

error rate 0.4% ↓

2. AlexNet

Performance details

(1) Two regularization approaches – reduce overfitting!

a. Dropout (50 %)

2 Fully connected layers were applied dropout process.

b. Data augmentation

1) Rotating , Reversal.

2) Change original ILSVRC 256x256 pixel images to 224x224 pixel data randomly.

(e.g one image can make 2048 different images)

3) Change light condition.

$$I_{xy} = [I_{xy}^{R},\ I_{xy}^{G},\ I_{xy}^{B}]^{T} + [\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3][\alpha_1\lambda_1, \alpha_2\lambda_2, \alpha_3\lambda_3]^{T}$$

$$\alpha_i \sim N(0,\ 0.1)$$

Perform PCA analysis about RGB pixel values!

More than error rate 1% ↓

2. AlexNet

Performance details

(3) ReLU function

a. Improve learning performance (same performance as normalization!)
Working effect of lateral inhibition similarly biological neuron .

b. Local response normalization
Used in the 1st, 2nd convolution layers after.

Equation 13-2. Local response normalization

$$b_i = a_i\left(k + \alpha \sum_{j=j_{\text{low}}}^{j_{\text{high}}} {a_j}^2\right)^{-\beta} \quad \text{with} \begin{cases} j_{\text{high}} = \min\left(i + \frac{r}{2}, f_n - 1\right) \\ j_{\text{low}} = \max\left(0, i - \frac{r}{2}\right) \end{cases}$$

$\alpha, \beta, k\ r$: *hyperparameter*

- b_i is the normalized output of the neuron located in feature map i, at some row u and column v (note that in this equation we consider only neurons located at this row and column, so u and v are not shown).
- a_i is the activation of that neuron after the ReLU step, but before normalization.
- k, α, β, and r are hyperparameters. k is called the *bias*, and r is called the *depth radius*.
- f_n is the number of feature maps.

3. GoogLeNet (Inception V1)

- Won ImageNet ILSVRC challenge 2014 (Christian Szegedy)
- More deeper than previous CNNs
- NetworkInNetwork(NIN) – *Inception modules*
- Use parameter more efficiently

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

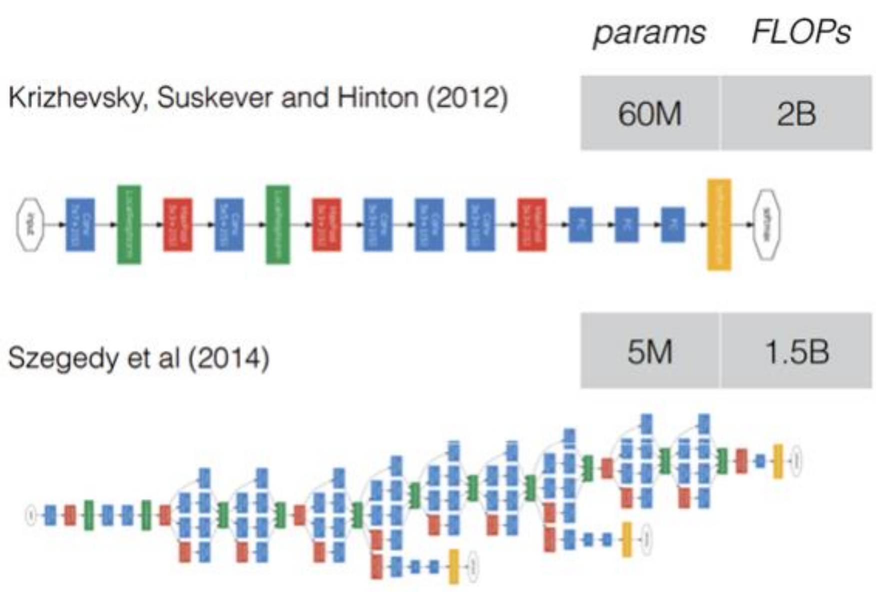
22 layers!

source : Going Deeper with Convoultions(CVPR 2015) – Christian Szegedy

3. GoogLeNet (Inception V1)

Performance details

(1) More deeper but lower parameter than previous CNNs



AlexNet

Convolution layer : 5
Max pooling : 2
Activation function : ReLU
Freeparameter : 60,000,000

GoogLeNet

1/10

Convolution layer : 64
Pooling layer : 16
Activation function : ReLU
Freeparameter : 6,000,000

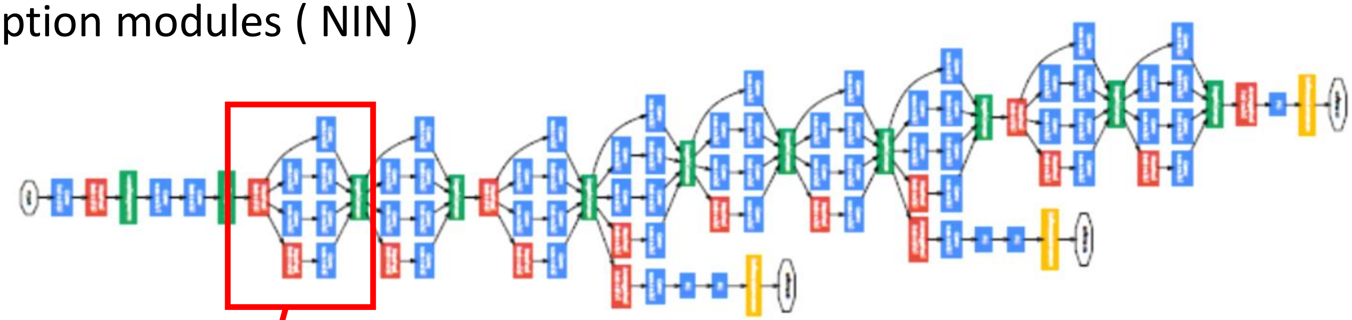
More deeper structure but Computional performance Better!

source : Going Deeper with Convoultions(CVPR 2015) – Christian Szegedy

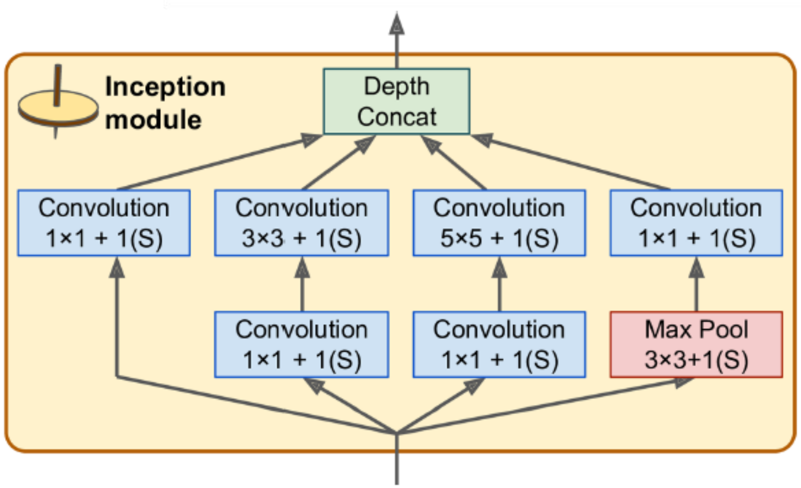
3. GoogLeNet (Inception V1)

Performance details

(2) Inception modules (NIN)



One Inception module obtains multiple networks



1x1 Convolutional layers : 4
3x3 Convolutional layers : 1
5x5 Convolutional layers : 1
3x3 Pooling layer : 1

1x1 Convolutional layer ??

source : Going Deeper with Convoultions(CVPR 2015) – Christian Szegedy

3. GoogLeNet (Inception V1)

Performance details

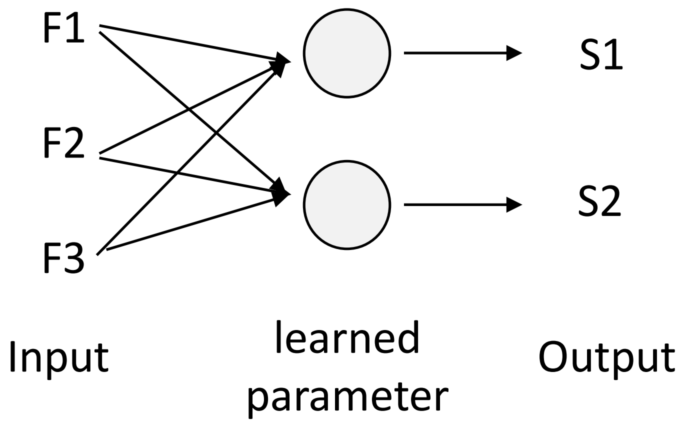
(2) Inception modules (NIN)

1x1 Convolutional layer

a. 1x1 convolutional layer are configured to output maps fewer feature maps than their input.

grouping several similar feature maps
Can **reduce the number of feature maps**.

ex) if the number of input feature maps are 3 and the number of feature maps are 2, they can be decided by 2 feature maps by learned parameter



Reduce dimensionality!

source : Going Deeper with Convoultions(CVPR 2015) – Christian Szegedy

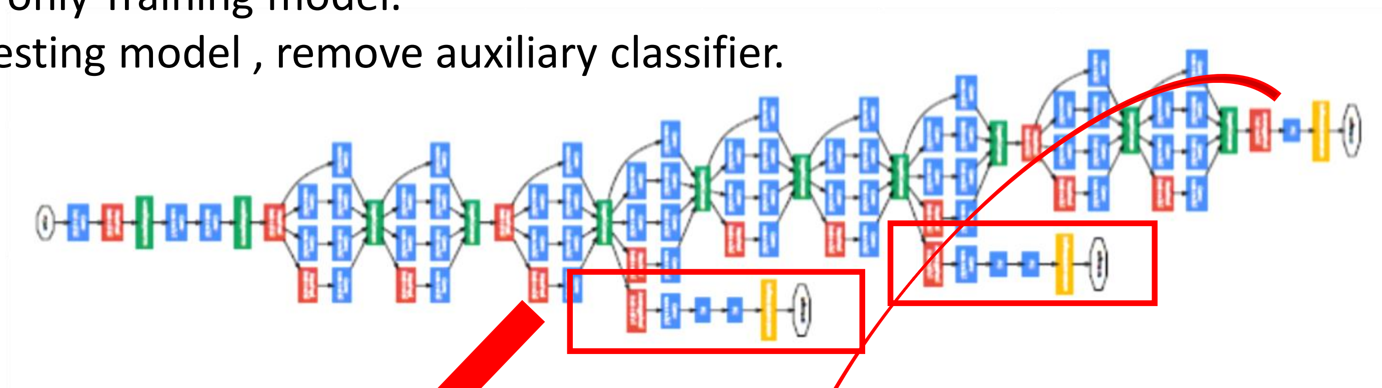
3. GoogLeNet (Inception V1)

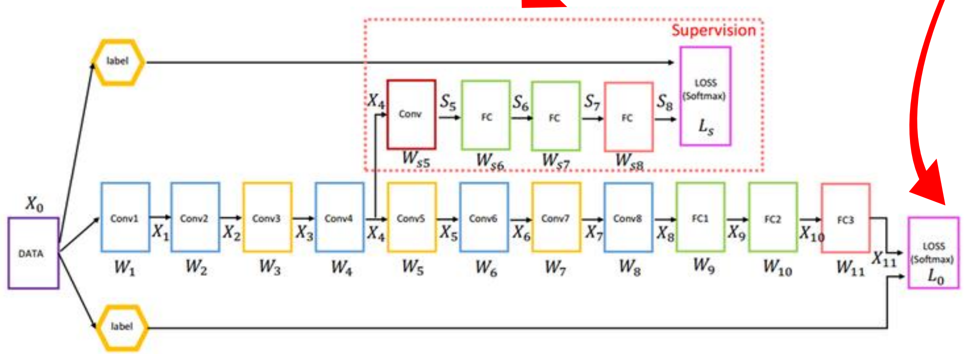
Performance details

(3) Auxiliary classifier

Use only Training model.

In Testing model , remove auxiliary classifier.





Solution of Vanishing Gradient problem
Improve Learning convergence

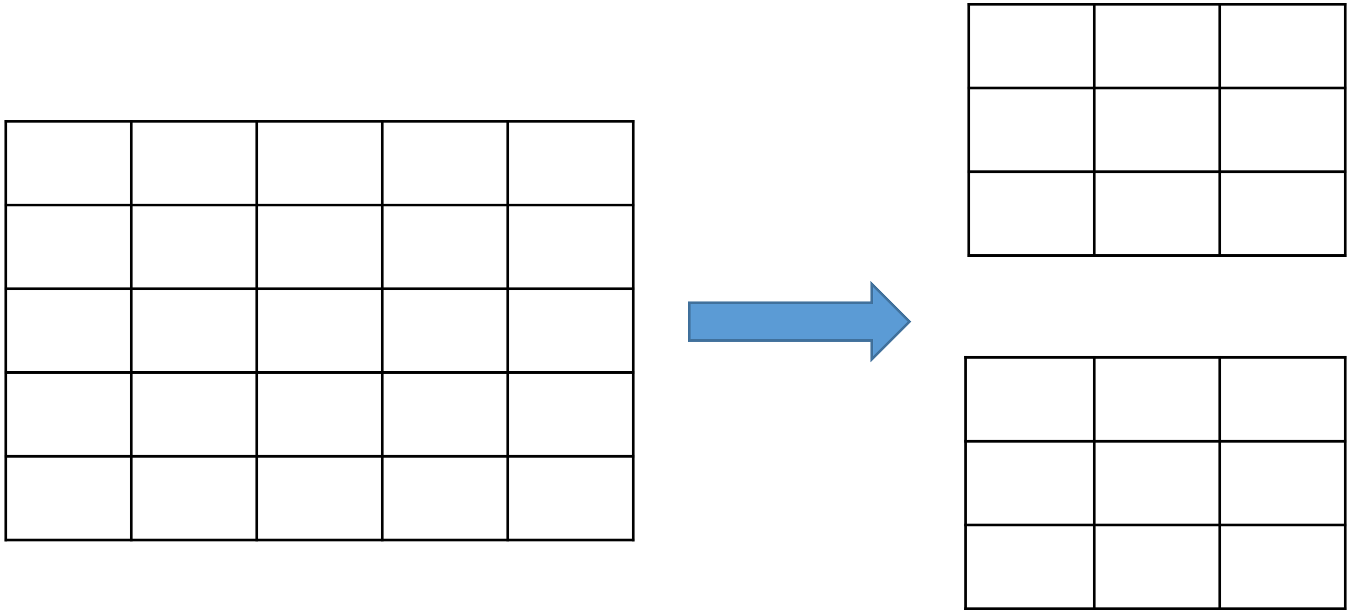
source : Going Deeper with Convoultions(CVPR 2015) – Christian Szegedy
Training Deeper Convolutional Networks with Deep SuperVision(2015) – Liwei Wang

3. GoogLeNet (Inception V1)

Performance details

(3) Factorizing convolutions

Factorizing big filter can be reduce free parameters.



5x5 filter (25 parameters)

2 3x3 filters (9+9 parameters)

source : Going Deeper with Convoultions(CVPR 2015) – Christian Szegedy
Training Deeper Convolutional Networks with Deep SuperVision(2015) – Liwei Wang

4. ResNet

- Won ImageNet ILSVRC challenge 2015 (kaiming He)

- 152 layers (extremely deep)

- Use skip connections(shortcut connections)

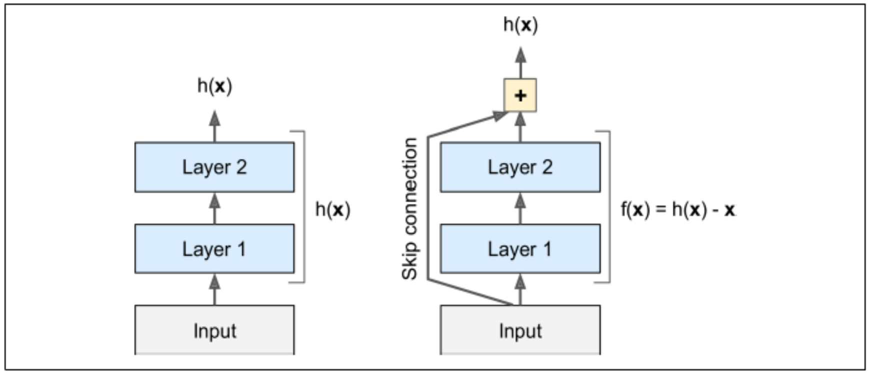


Figure 13-12. Residual learning

Target function is h(x)
not f(x) = h(x) – x.

When NN's weights are close to zero,
output values are almost 0.
Then, Skip connection is applied .

It solves learning performance of extremely deep networks!

source : Going Deeper with Convoultions(CVPR 2015) – Christian Szegedy
Training Deeper Convolutional Networks with Deep SuperVision(2015) – Liwei Wang

Thank you