

『Hands-On Machine Learning』

Chapter 13. Convolutional neural network

SangHyeok Kim

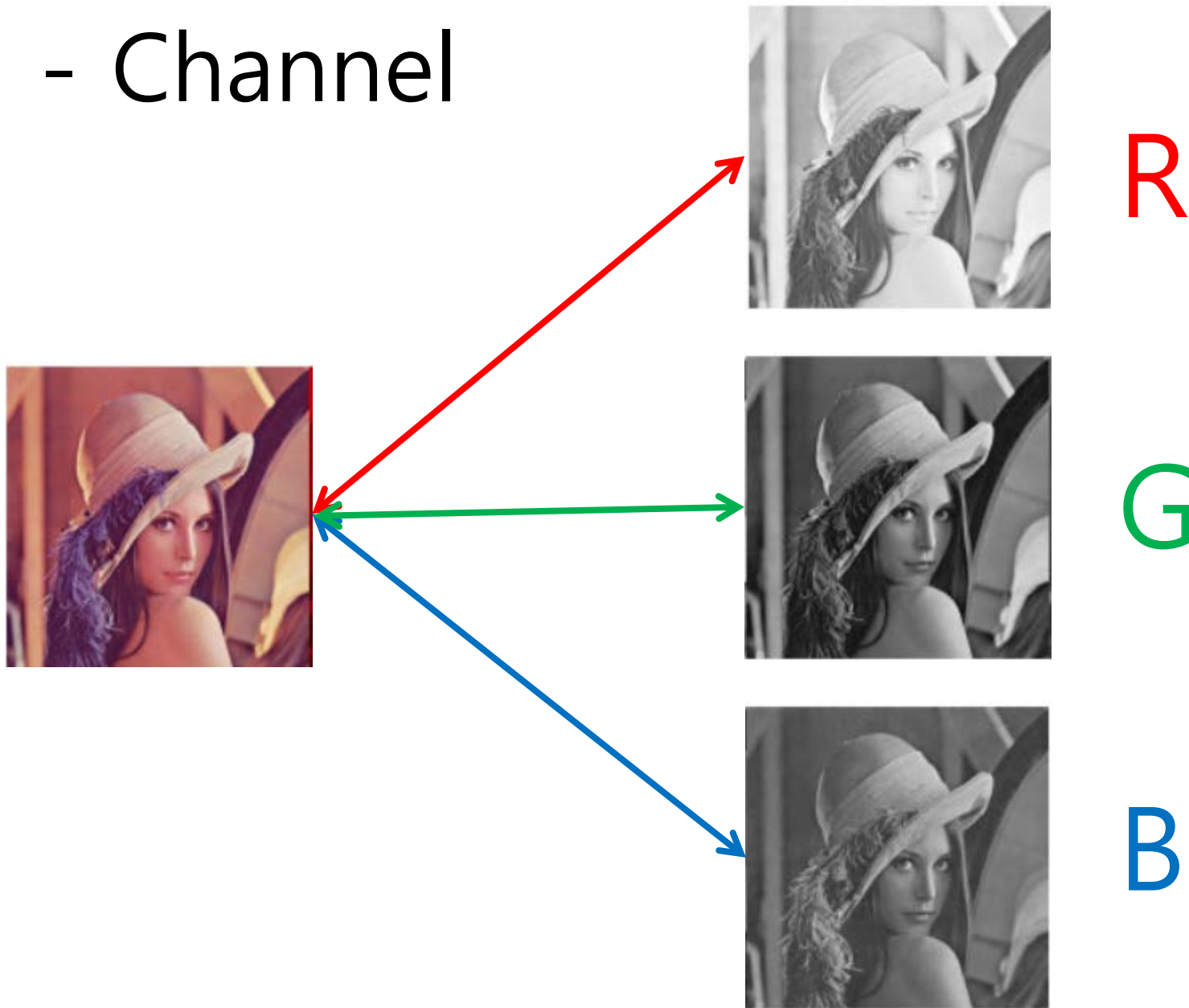
Contents

1. Convolutional Neural Network
2. 주요 용어 설명
3. CNN의 기본 구조
4. 다양한 CNN

1. Convolutional Neural Network란?

- CNN 또는 ConvNet라고 불리우며 대뇌의 시각 피질 연구로부터 시작되어 1980년대부터 이미지 인식 분야에 사용되었다.
- 컴퓨팅 성능의 향상과 축적된 데이터로 인해 이미지 처리 능력이 획기적으로 향상되었으며, 이미지 외에도 비디오나 텍스트 또는 사운드를 분류하는 데에 많이 사용된다.
- CNN은 이미지에서 객체나 얼굴, 장면을 인식하기 위한 패턴을 찾는 데에 유용하며 데이터에서 직접 학습하기에 특징을 수동으로 추출할 필요가 없다.

2. 용어 - Channel



2. 용어 - Filter, Convolution, Feature map

Input

1	0	0	0
1	1	1	1
0	1	0	0
1	1	0	1

Filter

1	0	0
0	1	1
1	1	1

Convolution

1	0	0	0	0
0	1	1	1	1
1	0	1	1	0
1	1	0	1	

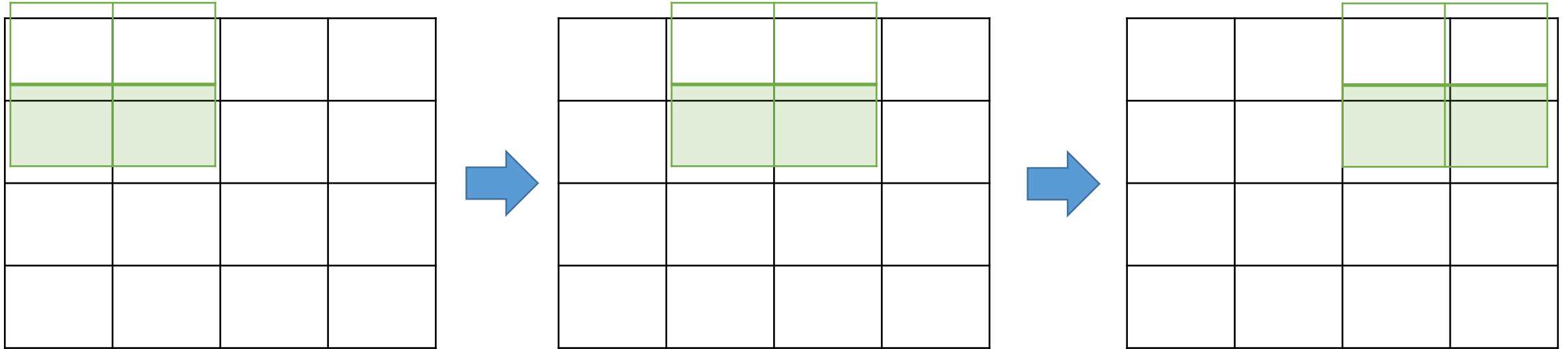
Feature Map

4		

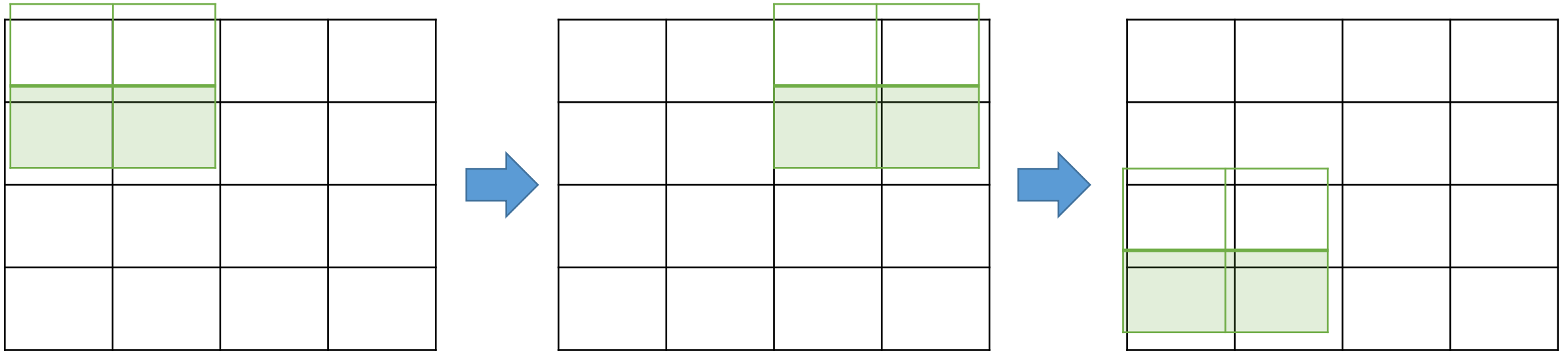


$$\begin{aligned} &1*1 + 0*0 + 0*0 + \\ &0*1 + 1*1 + 1*1 + \\ &1*0 + 1*1 + 1*0 = 4 \end{aligned}$$

2. 용어 - Strid 1



2. 용어 - Strid 2



2. 용어 - Padding

- Filter로 인한 Convolution작업과 Stride의 설정 등으로 인해 Feature Map의 크기는 기존의 Input Data보다 작아지게 된다. 이때, Convolution layer의 출력 데이터가 줄어드는 것을 방지하기 위해 사용하는 방법이다.
- 입력 데이터의 외각에 지정된 픽셀만큼 특정한 값으로 채워 넣는 것을 의미하며 주로 0으로 채워넣고 'Zero Padding'이라 한다.

2. 용어 - Zero Padding



2. 용어 - Pooling Layer

- 출력 데이터를 입력으로 받아 출력 데이터의 크기를 줄이거나 특정 데이터를 강조하는 용도로 사용된다.
- Max Pooling, Average Pooling, Min Pooling 등이 있으며, 주로 Pooling의 크기와 Stride를 같은 크기로 설정한다.

2. 용어 - Pooling Layer

6	15	21	31
3	28	23	65
41	5	56	12
55	87	12	4

Max pooling

28	65
87	56

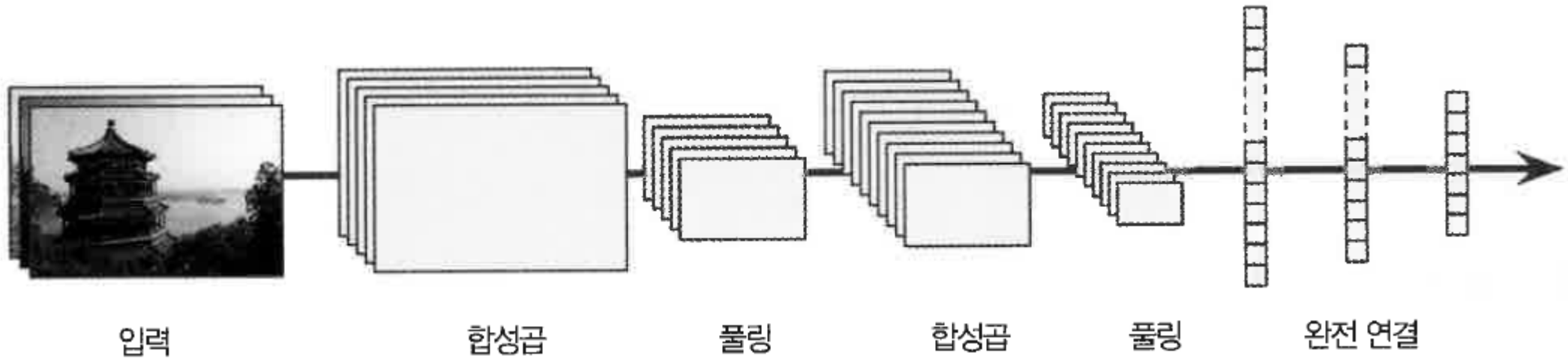
Min pooling

3	21
5	4

Average pooling

13	35
47	21

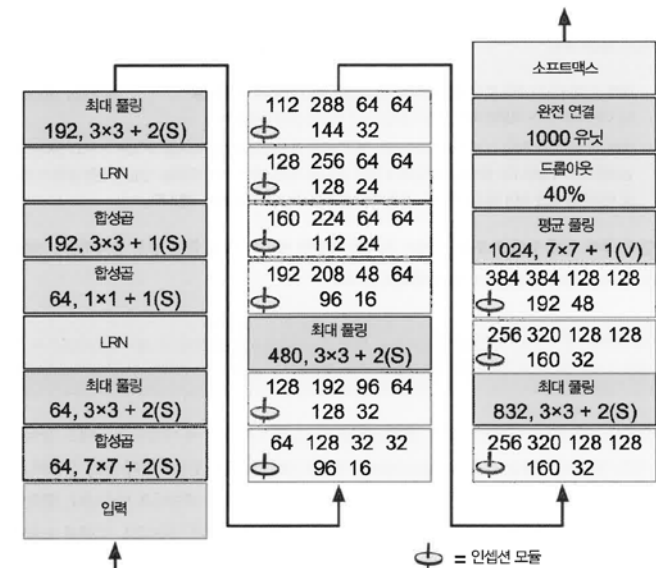
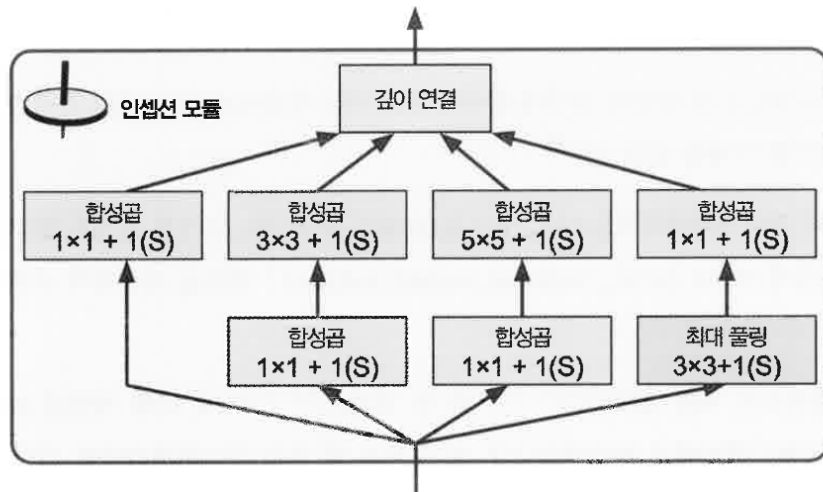
3. CNN의 구조



- 일반적인 CNN의 구조는 몇 개의 합성곱층을 쌓고 그 다음에 풀링층을 쌓고 다시 합성곱층을 쌓고 풀링층을 쌓는 방식으로 이루어져 있다.
- 진행될수록 이미지는 점점 작아지지만 합성곱층으로 인해 더 많은 특성 맵을 갖게 된다.

4. CNN예시 – GoogLeNet(Going Deeper with Convolutions)

- 이전 CNN들에 비해 훨씬 더 깊은 네트워크를 이용하여 에러율을 낮추었다.
- 인셉션 모듈이라는 서브 네트워크를 통해 GoogLeNet이전의 구조보다 훨씬 효과적으로 파라미터를 사용하며 AlexNet보다 10배 적은 파라미터를 가진다.



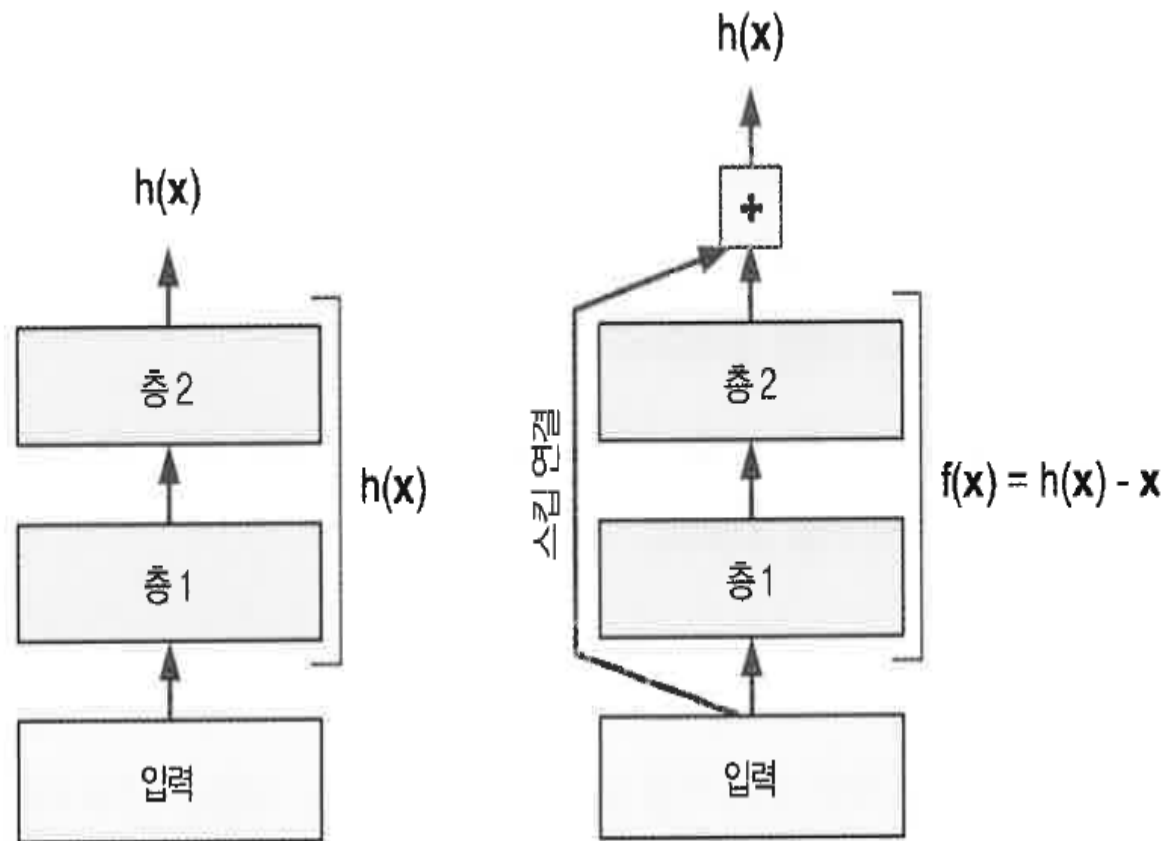
4. GoogLeNet

- 일반적인 CNN에서는 네트워크가 깊어질수록(더 많은 레이어를 가질수록) 성능이 좋아진다. 하지만, 네트워크가 깊어질수록 학습해야 할 파라미터 수가 늘어난다.
- 이미지를 예시로 들면 CNN에서 convolution 연산을 할 시 channel은 커지고 height와 width는 줄어들게 되는데, 예를 들어 192x28x28의 크기의 데이터를 5x5 필터, stride-1, padding=2로 총 32개의 필터를 동작시키면 총 필요 연산이 $192 \times 5 \times 5 \times 32 \times 28 \times 28 = \text{약 } 120\text{만}$ 개가 되어 여러 층을 쌓기 부담스럽게 된다.
- GoogLeNet을 사용하면 이러한 문제점을 어느 정도 해결할 수 있다.

4. CNN예시 - ResNet

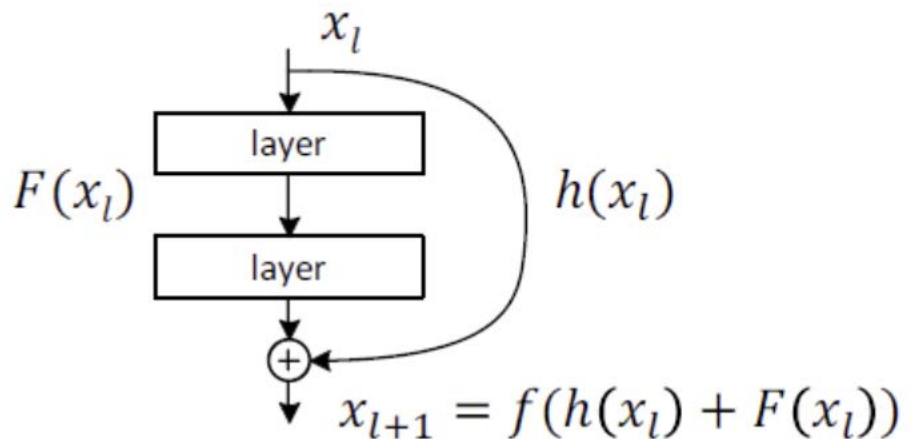
- 152개 층으로 구성된 깊은 CNN을 사용하여 3.6%이하의 에러율을 기록함.
- 어떤 층에 주입되는 신호가 상위 층의 출력에도 더해지는 스킵 연결(숏컷 연결)을 통해 네트워크를 훈련시켰다.

4. ResNet - Residual Network

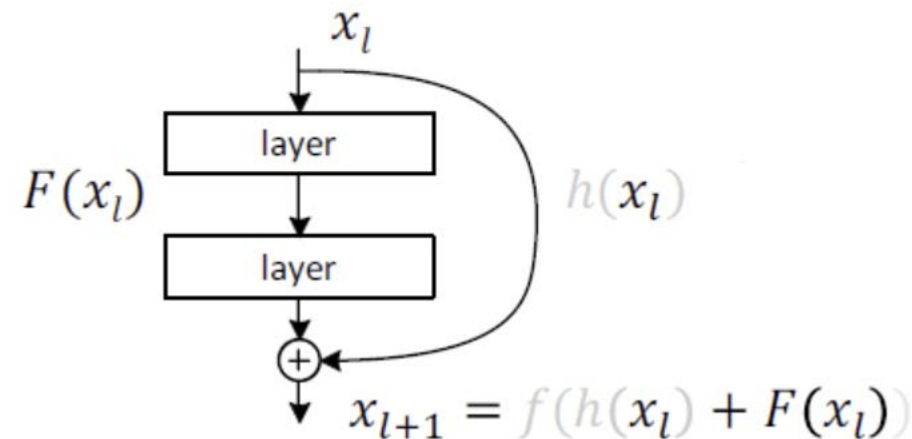


- 스킵 연결을 추가하여 입력과 같은 값을 출력하도록 하여 훈련 속도를 올리고, 스킵 연결을 많이 추가하여 일부 층이 학습되지 않았더라도 네트워크가 훈련을 시작할 수 있도록 한다. -> 입력 신호가 전체 네트워크에 쉽게 영향을 미친다.

4. ResNet – Residual Network



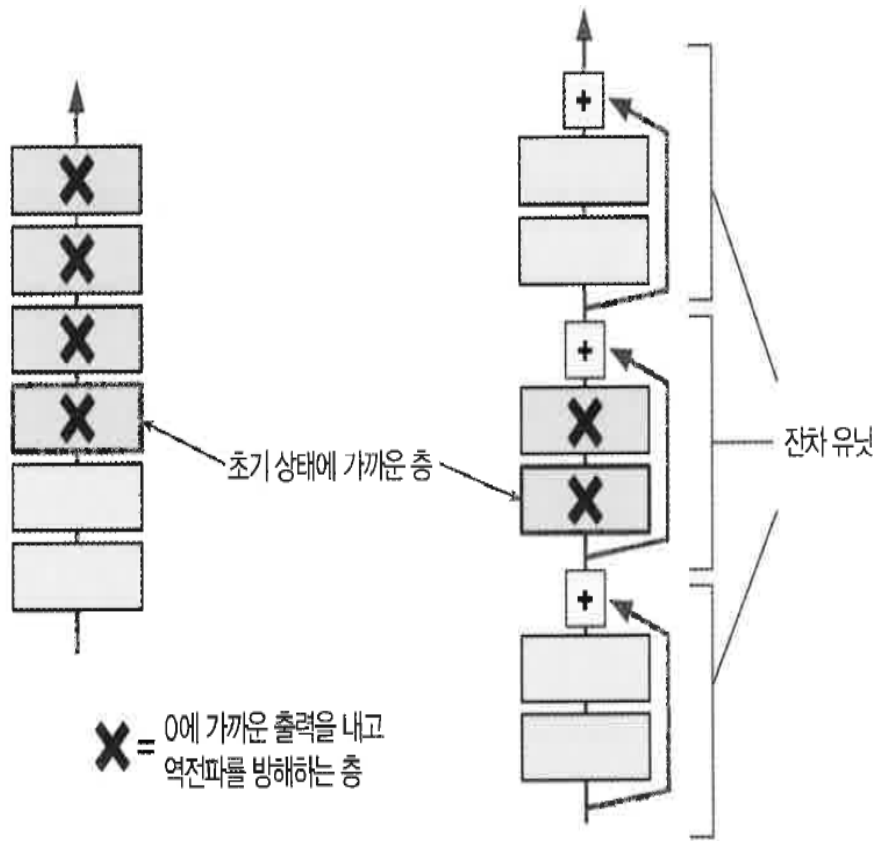
기존의 Residual Network



변경된 Residual Network

- 일반적인 ConvNet의 경우에는 forward나 backward path가 곱의 형태로 표현되는데 위와 같이 변경된 Residual Network를 사용하면 임의의 위치에서의 x_l 을 x_l 과 residual함수의 합으로 표현 가능하기에 forward나 backward path를 아주 간단하게 표현할 수 있게 된다.

4. ResNet - ResNet의 형태



- ResNet은 스킵 연결을 가진 작은 신경망인 Residual Unit을 쌓은 형태로 이루어져 있다.
- 그 외의 형태는 단순하며 GoogLeNet과 똑같이 시작되고 종료된다.

End