Machine & Deep learning basics [AICS305]
Machine learning for cyber security (CNN)

Prof. Mee Lan Han (aeternus1203@gmail.com)

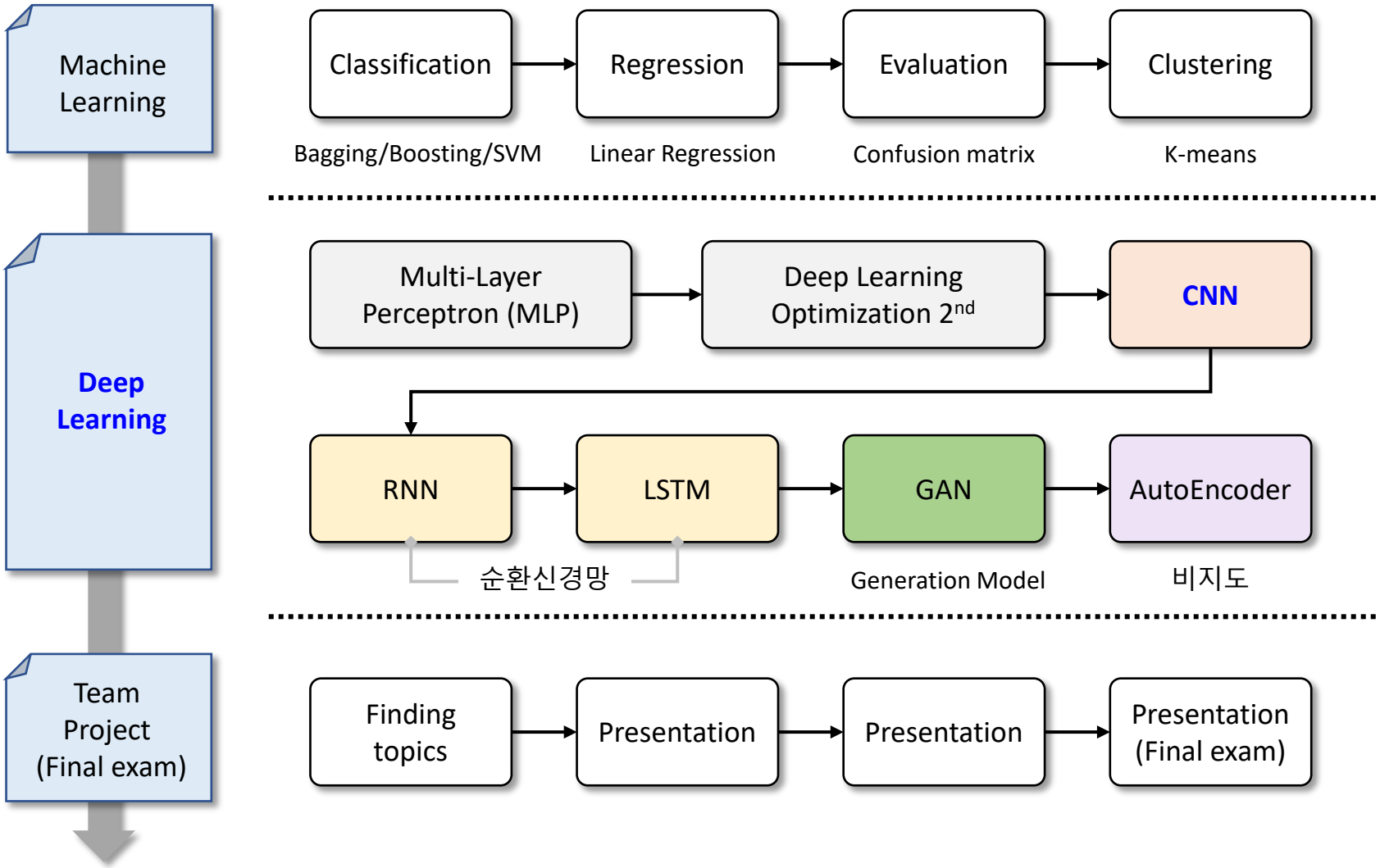
고려대학교

인공지능사이버보안학과

KOREA UNIVERSITY

Machine Learning vs. Deep Learning

■Study Plan



CONTENTS

- Convolutional Neural Network

Convolutional Neural Network (CNN)

■목차

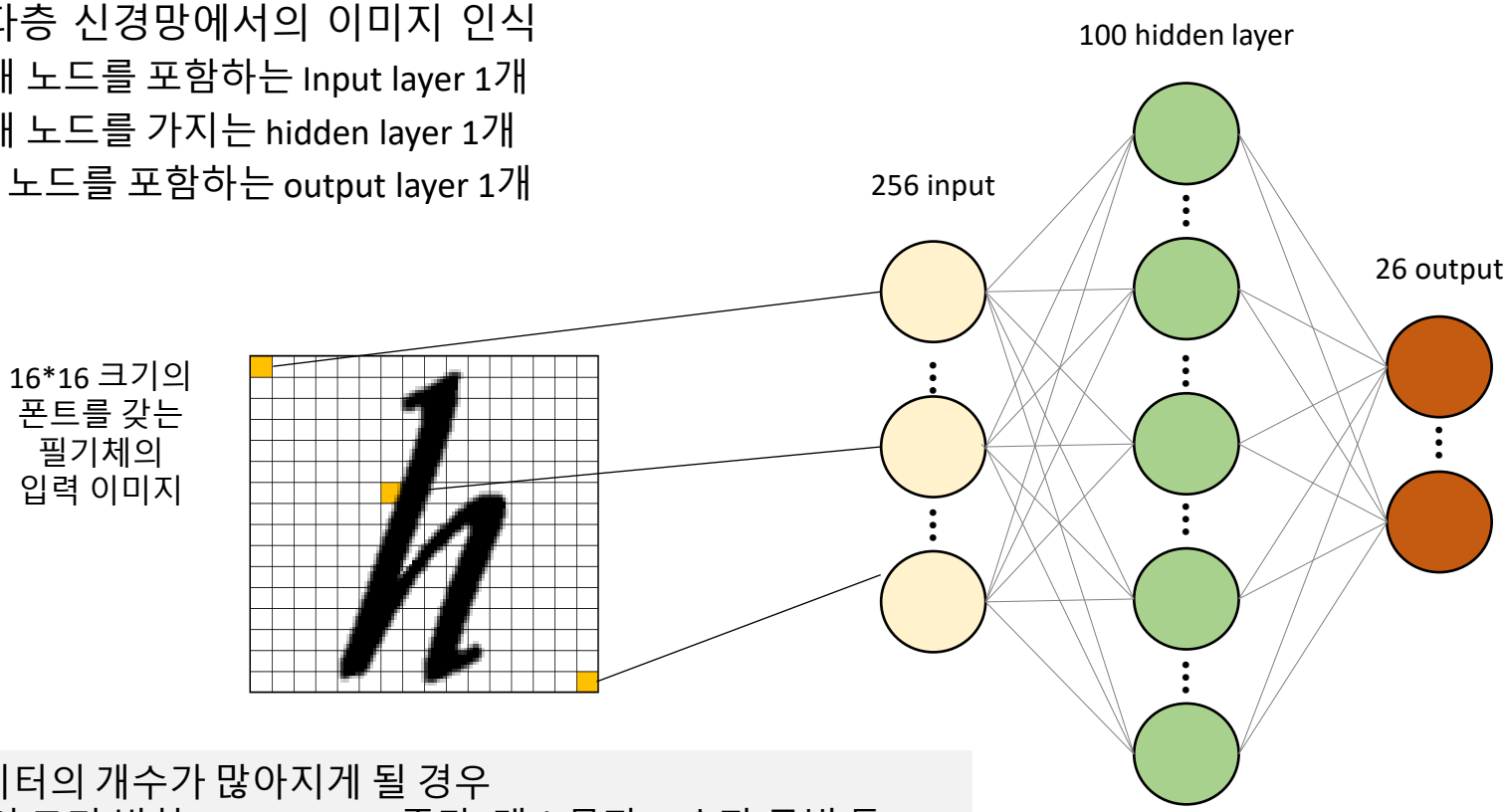
Convolutional Neural Network (CNN)

■ CNN 개요

□ 다층 신경망: 영상 (이미지) 데이터 기반의 인식 알고리즘 적용의 한계 존재

□ 기존 다층 신경망에서의 이미지 인식

- 256개 노드를 포함하는 Input layer 1개
- 100개 노드를 가지는 hidden layer 1개
- 26개 노드를 포함하는 output layer 1개



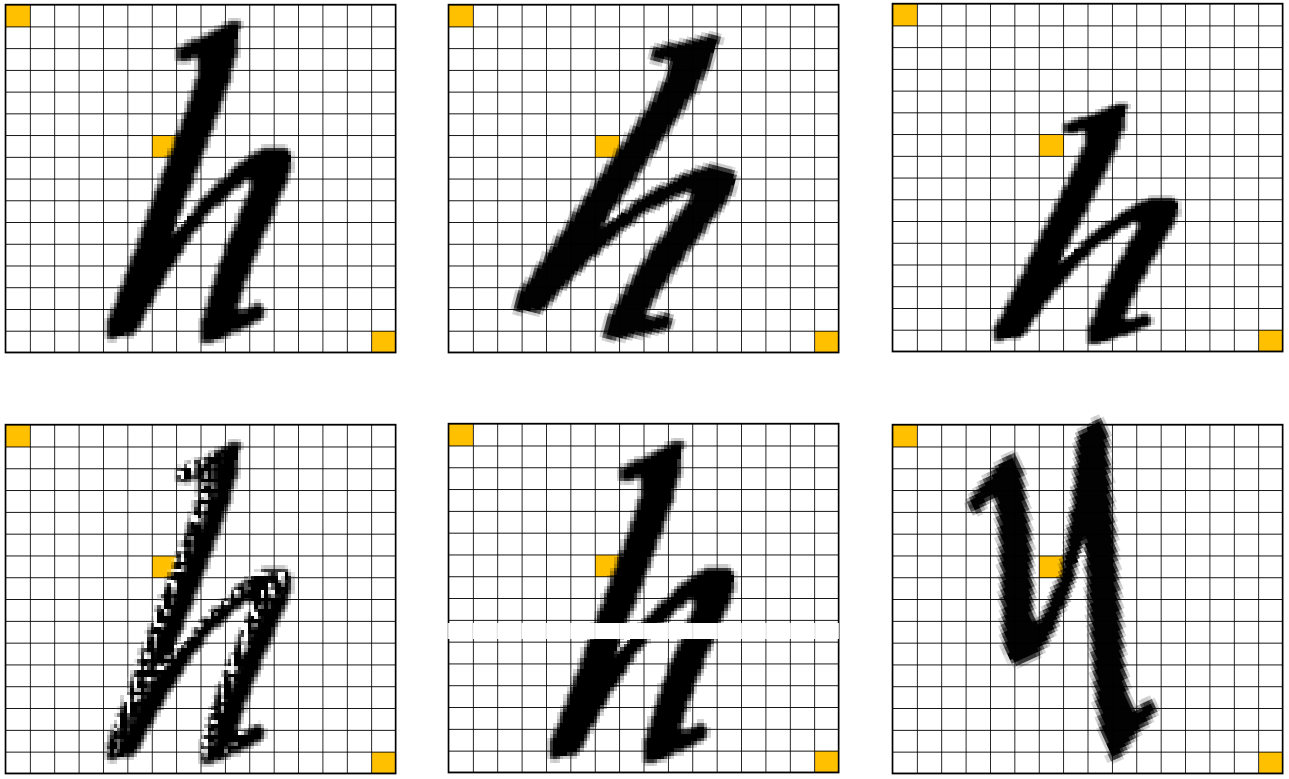
- 파라미터의 개수가 많아지게 될 경우
  (폰트의 크기 변화, hidden layer 증가, 대소문자 & 숫자 구별 등)
  학습 데이터로의 이미지 인식 처리의 문제 발생

Convolutional Neural Network (CNN)

■ CNN 개요

□ 기존 다층 신경망

- Image Topology (Size, Spin & Angle, Shape, Color 등) 를 고려하지 않음
- **이미지 변형** 조금이라도 생기게 될 경우 해당 데이터로의 **학습 없이는 결과가 좋지 않음**

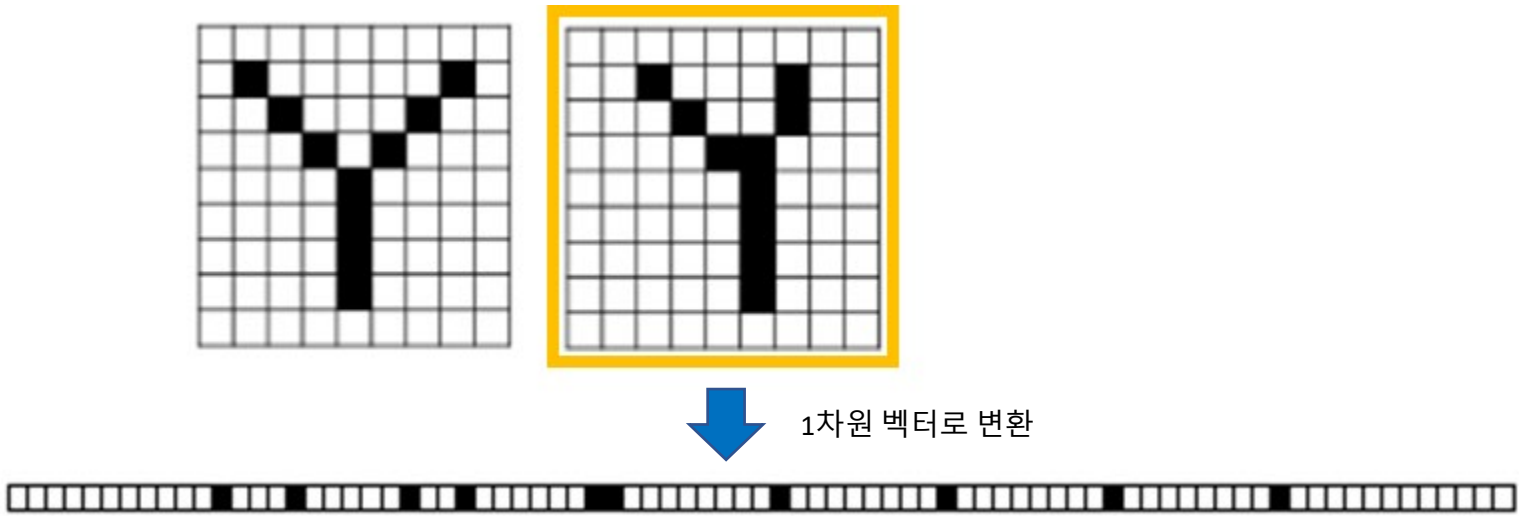


Convolutional Neural Network (CNN)

■ CNN 개요

□ 기존 다층 신경망

- Y 라는 글자 이미지를 다층 퍼셉트론으로 분류
- 이미지를 1차원 텐서인 벡터로 변환하고 다층 퍼셉트론의 입력층으로 사용

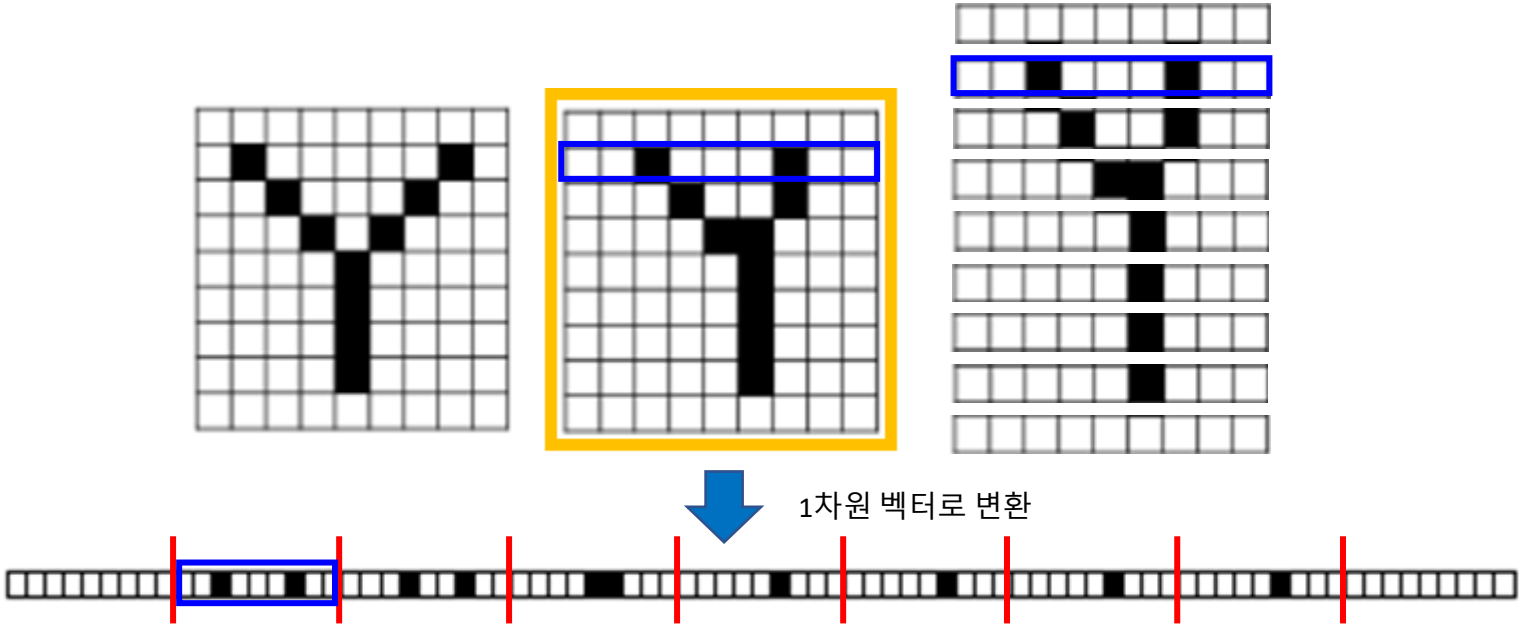


- 1차원 벡터로 변환 전 가지고 있던 공간적인 구조 정보가 유실됨
- 공간적인 구조 정보
  - 거리가 가까운 픽셀들 간의 값의 연관성
  - 어떤 픽셀의 값은 비슷하거나 다른 경우들이 존재함
- 공간적인 구조 정보를 보존하면서 이미지 데이터를 학습할 방법이 요구됨

Convolutional Neural Network (CNN)

■ CNN 개요

□ 기존 다층 신경망



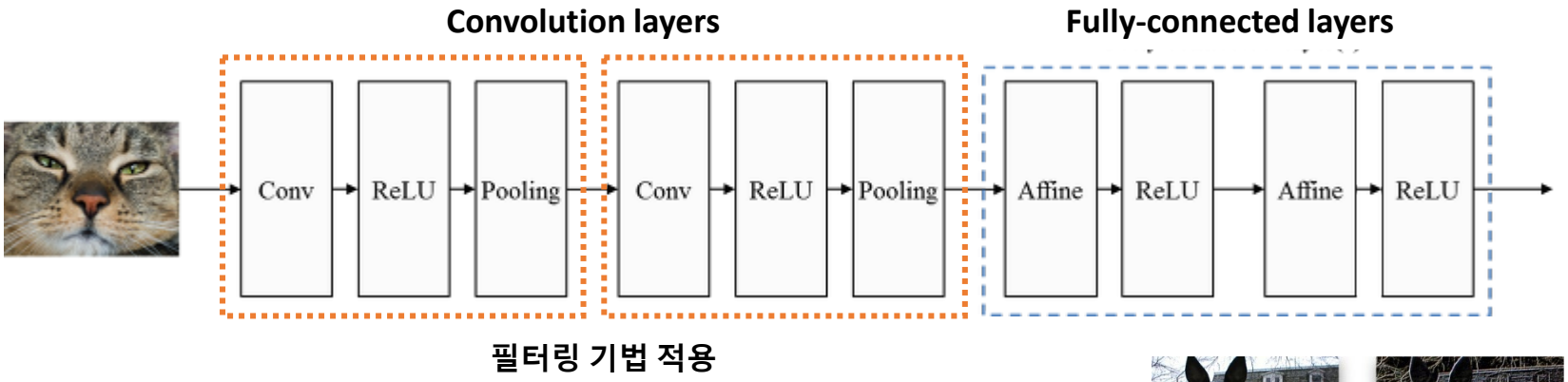
- 1차원 벡터로 변환 전 가지고 있던 공간적인 구조 정보가 유실됨
- 공간적인 구조 정보
  - 거리가 가까운 픽셀들 간의 값의 연관성
  - 어떤 픽셀의 값은 비슷하거나 다른 경우들이 존재함
- 공간적인 구조 정보를 보존하면서 이미지 데이터를 학습할 방법이 요구됨

Convolutional Neural Network (CNN)

■ CNN 개요

□ 필터링 기법을 인공신경망에 적용하여 이미지를 더욱 효과적으로 처리하기 위한 알고리즘
□ 기존의 필터링 기법은 고정된 필터를 이용하여 이미지 처리
□ CNN은 행렬로 표현된 필터의 각 요소가 데이터 처리에 적합하도록 자동으로 학습되도록 함

□ 합성곱 계층과 풀링 계층이라고 하는 새로운 층을 fully-connected 계층 이전에 추가함
□ 원본 이미지에 필터링 기법을 적용한 뒤 필터링 된 이미지에 대해 분류 연산이 수행되도록 구성



□ **합성곱 계층**
- 이미지를 분류하기 위해 Filter를 통해 중요한 특징 정보를 추출하는 계층

□ **풀링 계층**
- 이미지의 국소적인 부분들을 하나의 대표적인 스칼라 값으로 변환 (Abstract feature 추출)

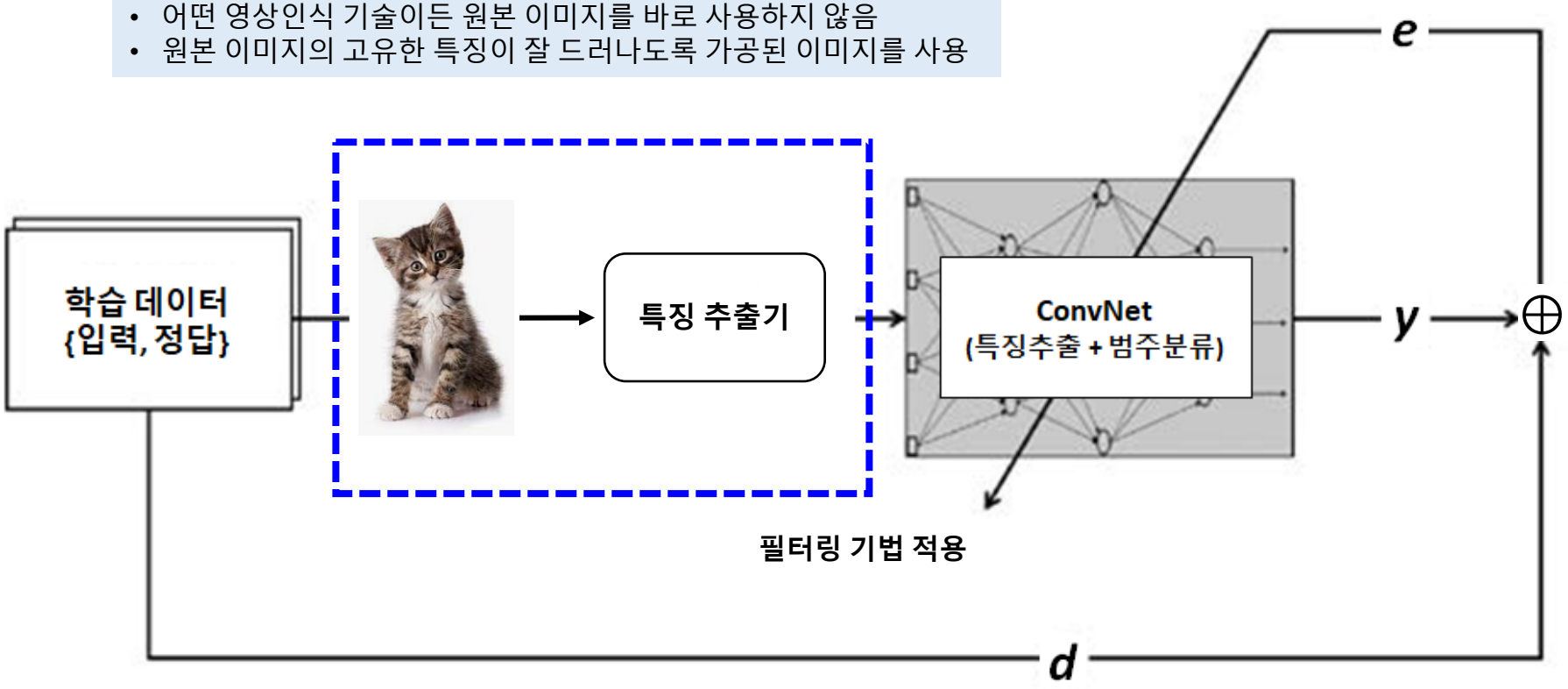
Convolutional Neural Network (CNN)

■ CNN 개요

□ 컨브넷 (ConvNet)

- 뇌의 시각피질의 이미지를 처리하고 인식하는 원리를 차용한 신경망
- 영상 또는 사진에서 고양이와 개가 어느 범주로 분류되는지 인식하는 문제!!
- **영상인식에서 사용되는 컨브넷의 출력층은 다범주 분류 (Multi-class Classification) 신경망**

  - 어떤 영상인식 기술이든 원본 이미지를 바로 사용하지 않음
  - 원본 이미지의 고유한 특징이 잘 드러나도록 가공된 이미지를 사용

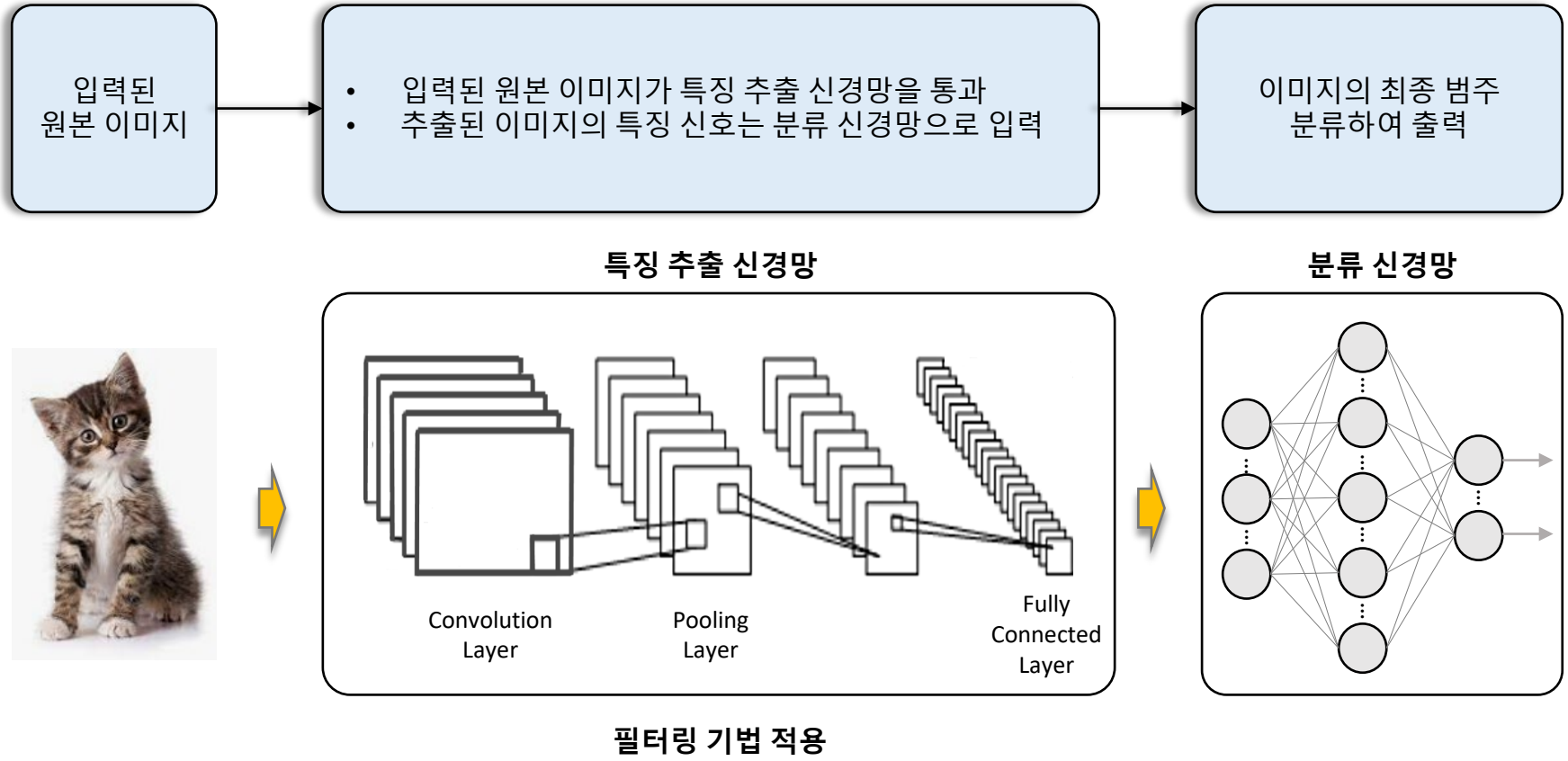


Convolutional Neural Network (CNN)

■ CNN 개요

□ 컨브넷 (ConvNet)

- 컨브넷은 특징 추출 신경망이 깊을수록 (층을 많이 쌓을수록) 영상 인식 성능도 좋아짐
- 신경망의 계층 구조가 깊어지면 학습 시키는데 어려움이 발생



Convolutional Neural Network (CNN)

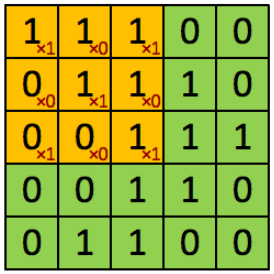
■ CNN Architecture

□ Convolution Layer (합성곱 계층)

- 입력 이미지 (데이터)로부터 합성곱 연산을 통해 중요한 정보를 추출해 내는 필터 역할을 수행
- **새로운 이미지,** 특징 맵 (**feature map**) 를 만들어내는 역할

- **어떤 합성곱 필터를 사용하느냐에 따라 합성곱 계층에서 추출해 내는 특징이 결정됨**

- 합성곱 계층은 일반적인 신경망의 계층과는 다른 구조와 방식으로 작동함
  - 합성곱 계층의 노드
    : 연결 가중치와 가중합의 개념이 아님
    : 입력 이미지를 다른 이미지로 변환하는 필터 (커널, kernel)로 이미지 처리

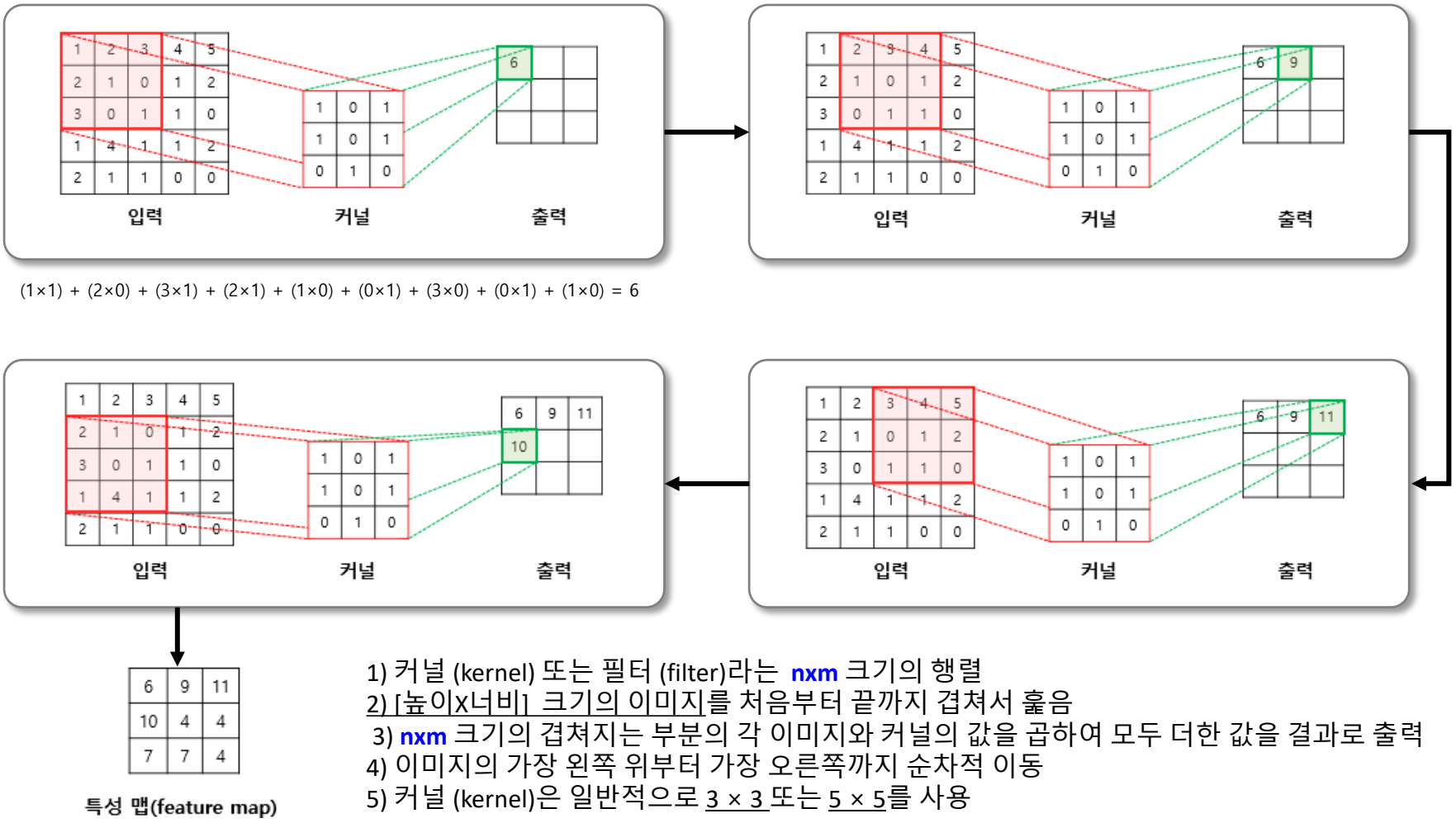


- Convolution Filter or Kernel (합성곱 필터 or 커널)
  - 입력 이미지를 다른 이미지로 변환하는 필터
  - 합성곱 필터로 입력 이미지를 처리하면 특징 맵을 얻을 수 있음

Convolutional Neural Network (CNN)

■ CNN Architecture

□ 3 × 3 크기의 커널로 5 × 5 이미지 행렬에 합성곱 연산을 수행하는 과정

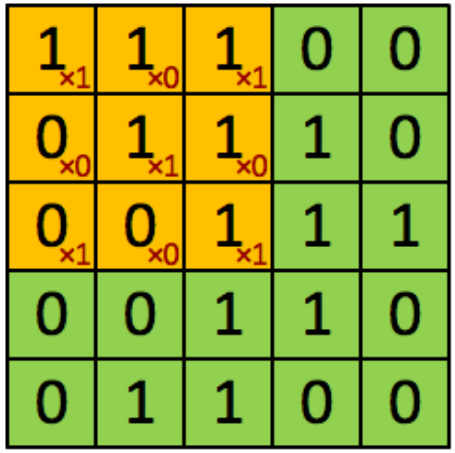


1) 커널 (kernel) 또는 필터 (filter)라는 **nxm** 크기의 행렬
2) [높이X너비] 크기의 이미지를 처음부터 끝까지 겹쳐서 훑음
3) **nxm** 크기의 겹쳐지는 부분의 각 이미지와 커널의 값을 곱하여 모두 더한 값을 결과로 출력
4) 이미지의 가장 왼쪽 위부터 가장 오른쪽까지 순차적 이동
5) 커널 (kernel)은 일반적으로 3 × 3 또는 5 × 5를 사용

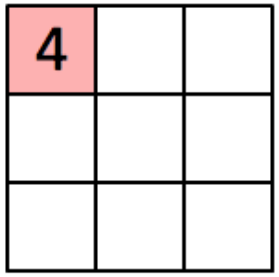
Convolutional Neural Network (CNN)

■ CNN Architecture

□ 3 × 3 크기의 커널로 5 × 5 이미지 행렬에 합성곱 연산을 수행하는 과정

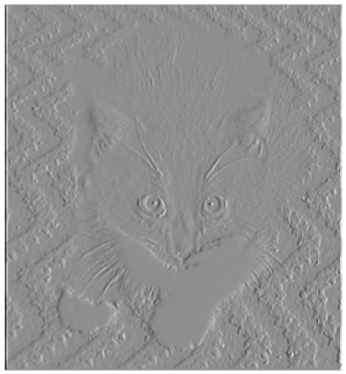


Image



Convolved Feature

Image

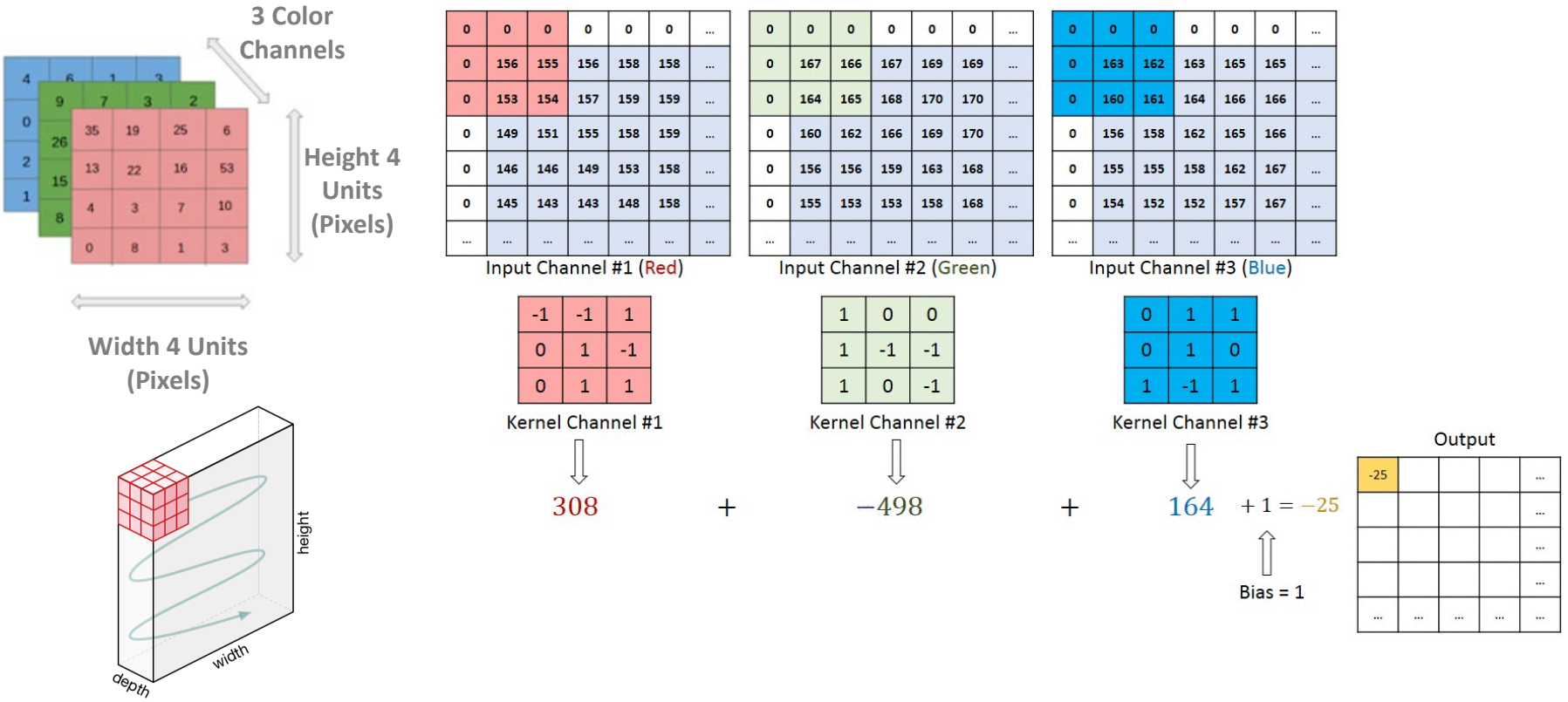


Convolved Feature

Convolutional Neural Network (CNN)

■ CNN Architecture

□ 3 × 3 크기의 커널로 5 × 5 이미지 행렬에 합성곱 연산을 수행하는 과정



[Ref] https://sonsnotation.blogspot.com/2020/11/7-convolutional-neural-networkcnn.html

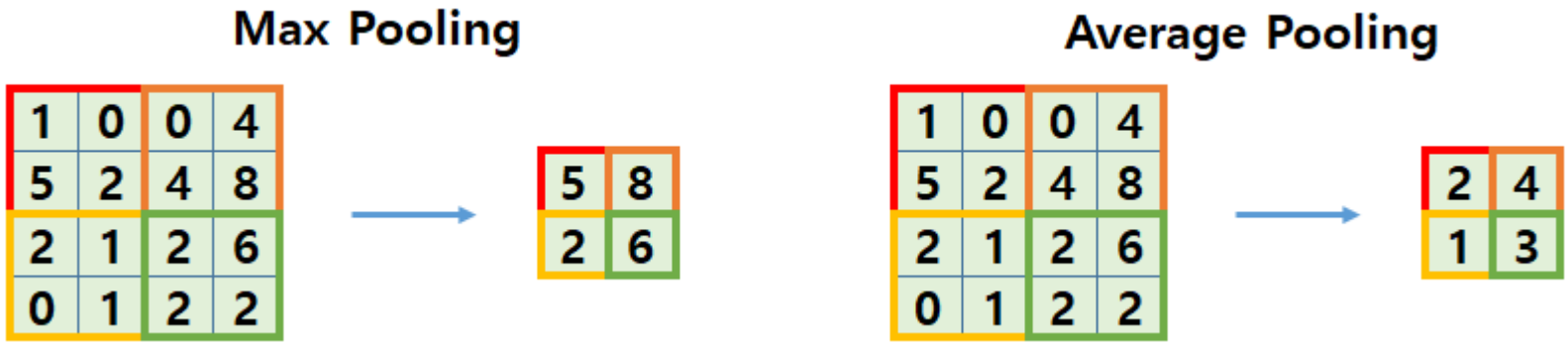
Convolutional Neural Network (CNN)

■ CNN Architecture

□ Pooling Layer (풀링 계층)

- 입력 이미지의 특정 영역에 있는 픽셀을 묶어서 하나의 대표 픽셀로 축소
- 즉, 이미지의 차원을 축소하여 이미지의 크기를 줄이는 역할을 함
- CNN 내 앞 layer의 출력 feature map의 모든 data가 필요하지 않기 때문에 pooling layer만 사용 (추론을 위한 적당량의 데이터만 있어도 충분함!!) -> overfitting ↓ speedup ↑

- **Max pooling**: pooling window 내의 가장 큰 값을 선택하는 방법
- **Average pooling**: 평균 연산으로 인해 학습 결과가 좋지 않음

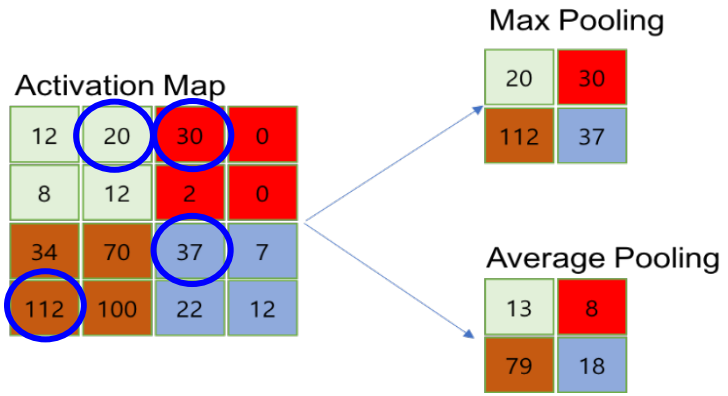


Convolutional Neural Network (CNN)

■ CNN Architecture

□ Pooling Layer (풀링 계층)

- sub-sampling을 이용하여 feature map의 크기를 줄이고, 위치나 이동에 강한 특징을 추출하기 위한 방법
- **Max pooling**: pooling window 내의 가장 큰 값을 선택하는 방법, Overfitting 되는 단점
- **Average pooling**: 평균 연산으로 인해 학습 결과가 좋지 않음



- **Stochastic pooling**
  - 최대값 또는 평균값 대신 확률에 따라 적절한 activation을 선택함
  - 확률값은 특정 activation에 대해 전체의 activation의 합을 나누는 방식으로 계산됨

$$p_i = \frac{a_i}{\sum_{k \in R_j} a_k}$$

  - Dropout과 같이 다양한 네트워크를 학습하는 듯한 model average 효과를 얻을 수 있음

Convolutional Neural Network (CNN)

■ CNN Architecture

□ Channel (채널)

- 이미지는 높이 (height), 너비 (width), 채널 (color) 이라는 3 Dimensional Tensor 구성
  - 높이: 이미지의 세로 방향 픽셀 수
  - 너비: 이미지의 가로 방향 픽셀 수
  - **채널: 색 성분**
    - 각 픽셀을 RGB 3개의 실수로 표현한 3차원 데이터
    - 컬러 이미지는 3개의 채널로 구성
    - 흑백 사진은 2차원 데이터로 1개 채널로 구성
    - 각 픽셀은 0부터 255 사이의 값으로 이루어짐

28*28 픽셀의 이미지
(28*28*1)의 3차원 텐서

□ 통상적으로 접하게 되는 컬러 이미지

Red Channel

Green Channel

Blue Channel

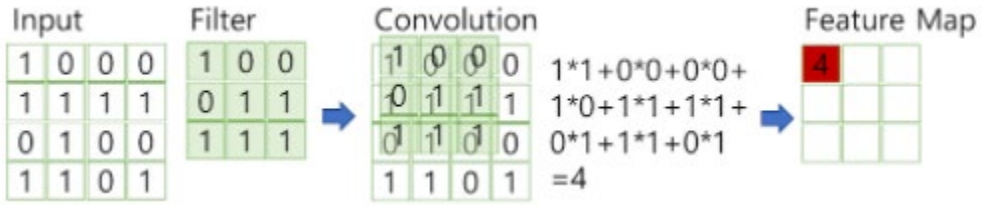
사막 이미지는 (28 × 28 × 3)의 크기를 가지는 **3차원 텐서**

Convolutional Neural Network (CNN)

■ CNN Architecture

□ Filter (=Kernel)

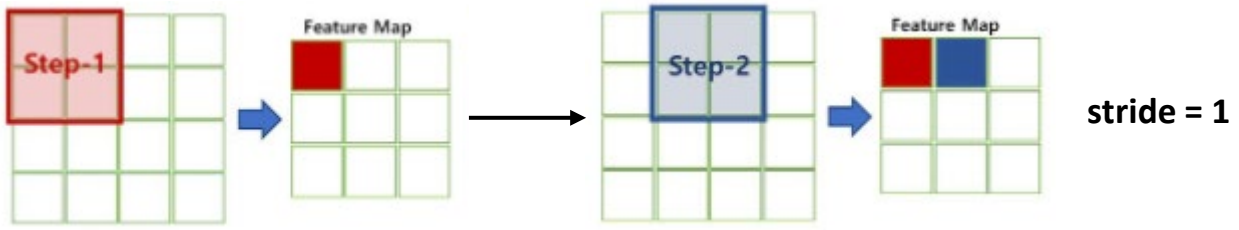
- 필터는 이미지의 특징을 찾아내기 위한 공용 파라미터
- 필터는 일반적으로 (4, 4)이나 (3, 3)과 같은 정사각 행렬로 정의
- CNN에서 학습의 대상은 필터 파라미터



- 입력 데이터를 지정된 간격으로 순회
- 채널별로 합성곱을 하고 모든 채널 (컬러의 경우 3개)의 합성곱의 합을 Feature Map로 만들어 냄

□ Stride

- 지정된 간격으로 필터를 순회하는 간격 (필터의 이동량)
- 필터는 입력 데이터를 지정한 간격으로 순회하면서 합성곱을 계산함

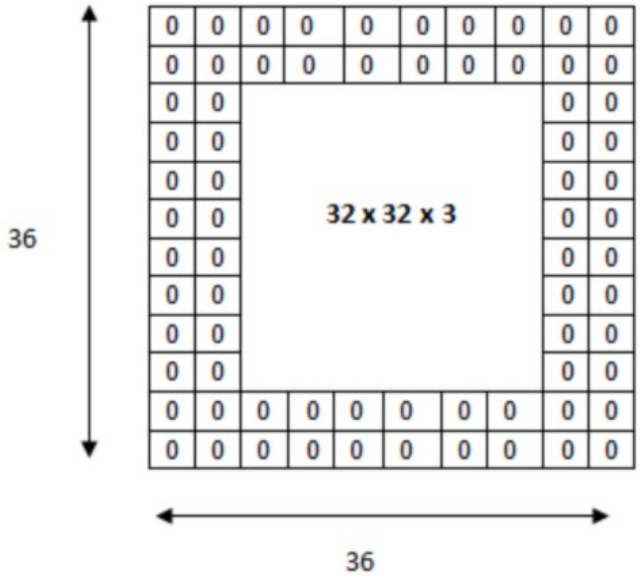


Convolutional Neural Network (CNN)

■ CNN Architecture

□ Padding

- 입력 데이터의 외각에 지정된 픽셀만큼 특정 값으로 채워 넣는 것을 의미
- Convolution 레이어의 출력 데이터가 줄어드는 것을 방지
- 보통 패딩 값으로 0으로 채움
- Convolution 레이어에서 Filter와 Stride에 작용으로 Feature Map 크기는 입력데이터 보다 작음



(32, 32, 3) 데이터를 외각에 2 pixel 추가 -> (36, 36, 3) 행렬

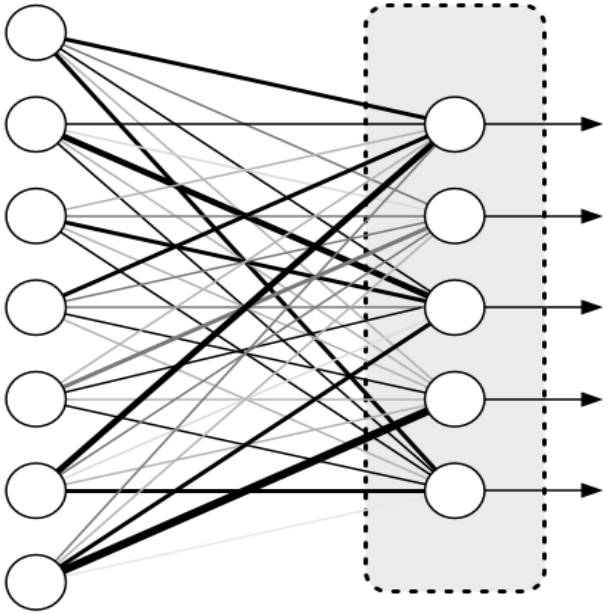
Convolutional Neural Network (CNN)

■ CNN Architecture

□ Fully Connected Layer (FCN): 평탄화

- Convolution Layer과 Pooling Layer으로부터 얻어진 특징 벡터들은 Fully Connected Layer의 입력으로 사용
- **이전 레이어의 출력을 평탄화 (Flatten)하여 다음 스테이지의 입력이 될 수 있는 단일 벡터로 변환**
- **비선형 공간에서의 분류를 수행하게 됨**

- 모든 뉴런들이 전부 연결되는 형태를 갖고 있기 때문에 Fully Connected 라는 이름으로 불림



Previous layer Fully connected layer

>> 1~3 과정을 Fully Connected Layers라고 정의함

① 2차원 배열 형태의 이미지를 1차원 배열로 평탄화
② 활성화 함수 (Relu, Tanh 등)뉴런을 활성화
③ 분류기 (Softmax) 함수로 분류

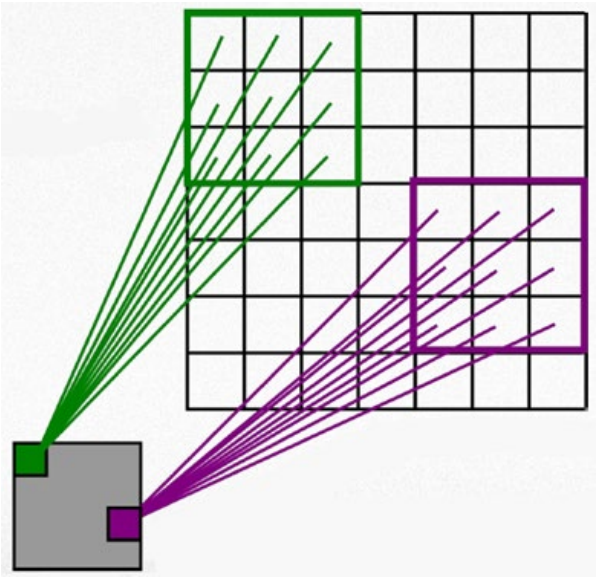
Convolutional Neural Network (CNN)

■ CNN 특징

□ CNN은 기존의 다층신경망에 비해 중요한 두 가지 특징이 존재 (Locality, Shared Weights)

□ **Locality (Local Connectivity)**

- CNN은 Receptive field (Kernel과 Stride)와 유사하게 local 정보를 활용함
- 공간적으로 인접한 신호들에 대한 correlation 관계를 비선형 필터를 적용하여 추출
- 비선형 필터를 여러 개 적용하면 다양한 local 특징 추출이 가능
- Sub-sampling 과정을 거치면서 영상의 크기는 줄어듦



Locally-connected units with 3x3 receptive field

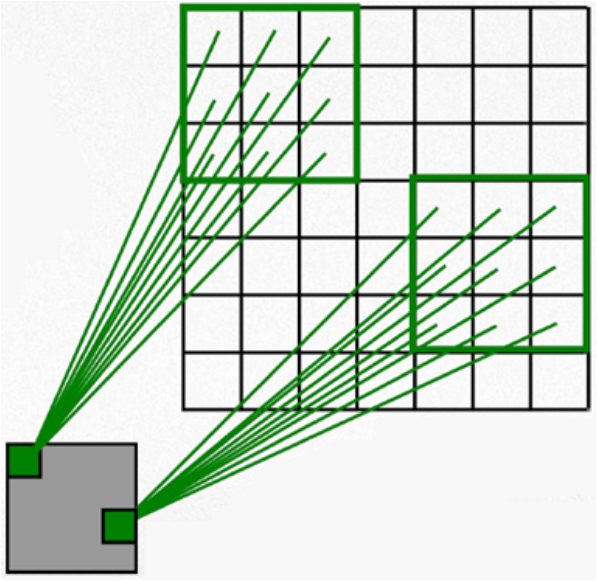
Convolutional Neural Network (CNN)

■ CNN 특징

□ CNN은 기존의 다층신경망에 비해 중요한 두 가지 특징이 존재 (Locality, Shared Weights)

□ **Shared Weights**

- 동일한 계수를 갖는 filter를 전체 영상에 반복적으로 적용함으로 변수의 수를 획기적으로 줄임
- Topology 변화에 무관하게 항상성 (Invariance) 을 얻을 수 있음



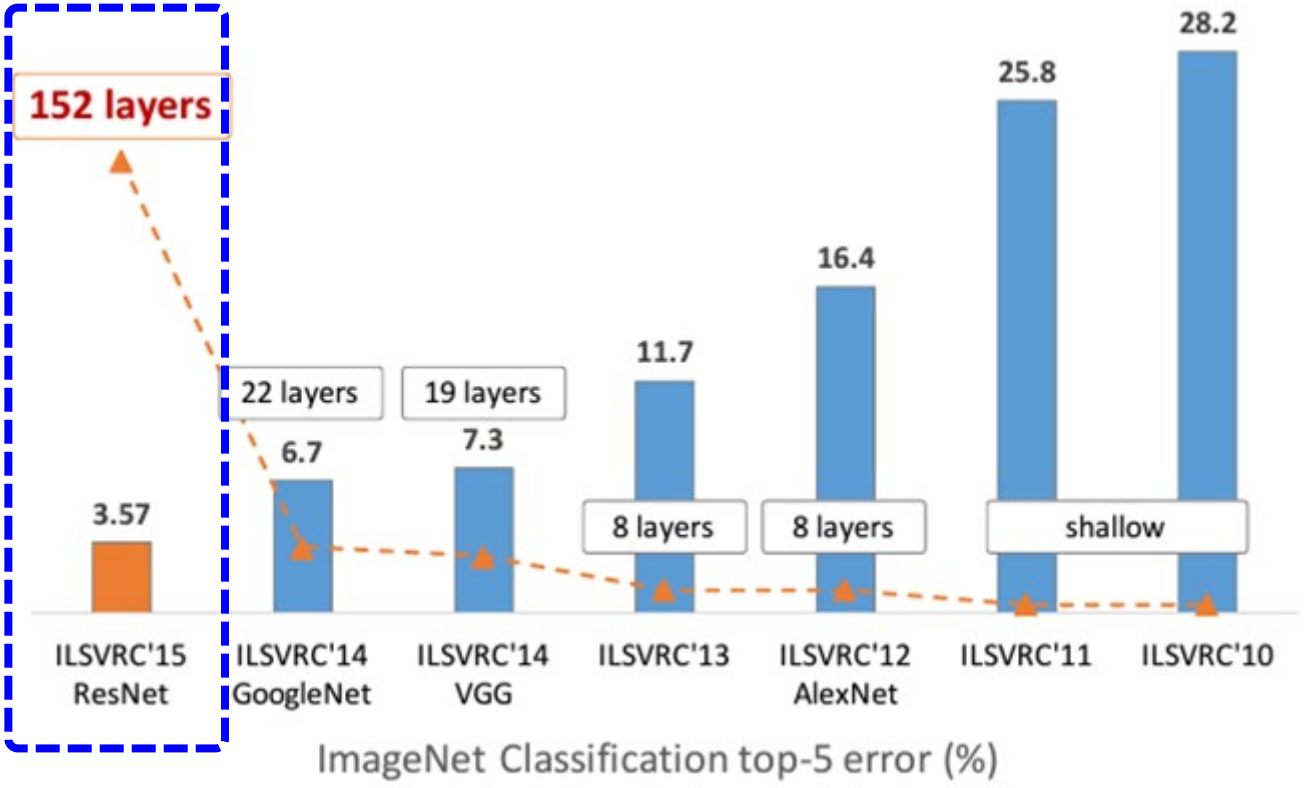
Convolutional units with 3x3 receptive field after sharing weight

Convolutional Neural Network (CNN)

■ CNN Architecture 모델

□ Revolution of Depth

- 2010년부터 2015년까지 이미지넷 대회 정확도 변화에 따라 나온 CNN 모델

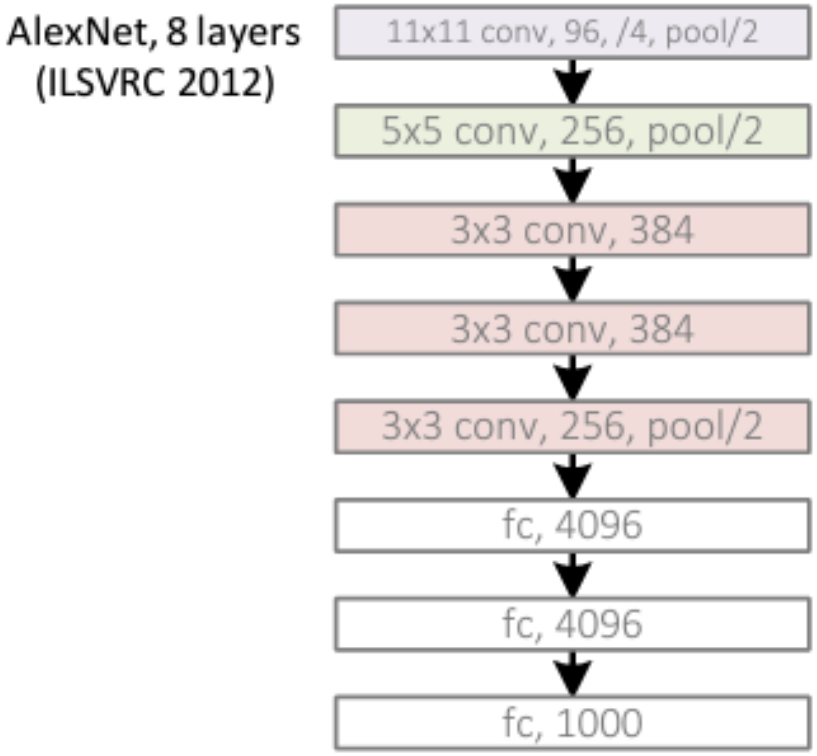


KaimingHe,XiangyuZhang,ShaoqingRen,&JianSun."DeepResidualLearningforImageRecognition".CVPR2016

Convolutional Neural Network (CNN)

■ CNN Architecture 모델

□ Revolution of Depth



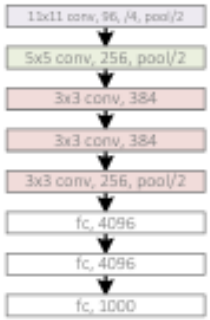
KaimingHe,XiangyuZhang,ShaoqingRen,&JianSun."DeepResidualLearningforImageRecognition".CVPR2016

Convolutional Neural Network (CNN)

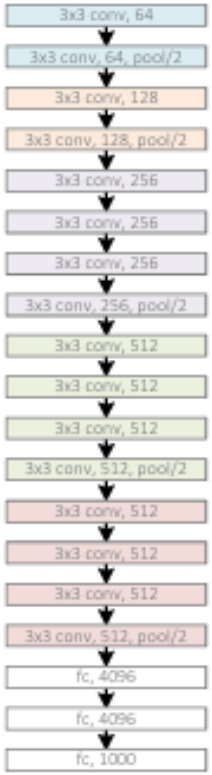
■ CNN Architecture 모델

□ Revolution of Depth

AlexNet, 8 layers
(ILSVRC 2012)



VGG, 19 layers
(ILSVRC 2014)



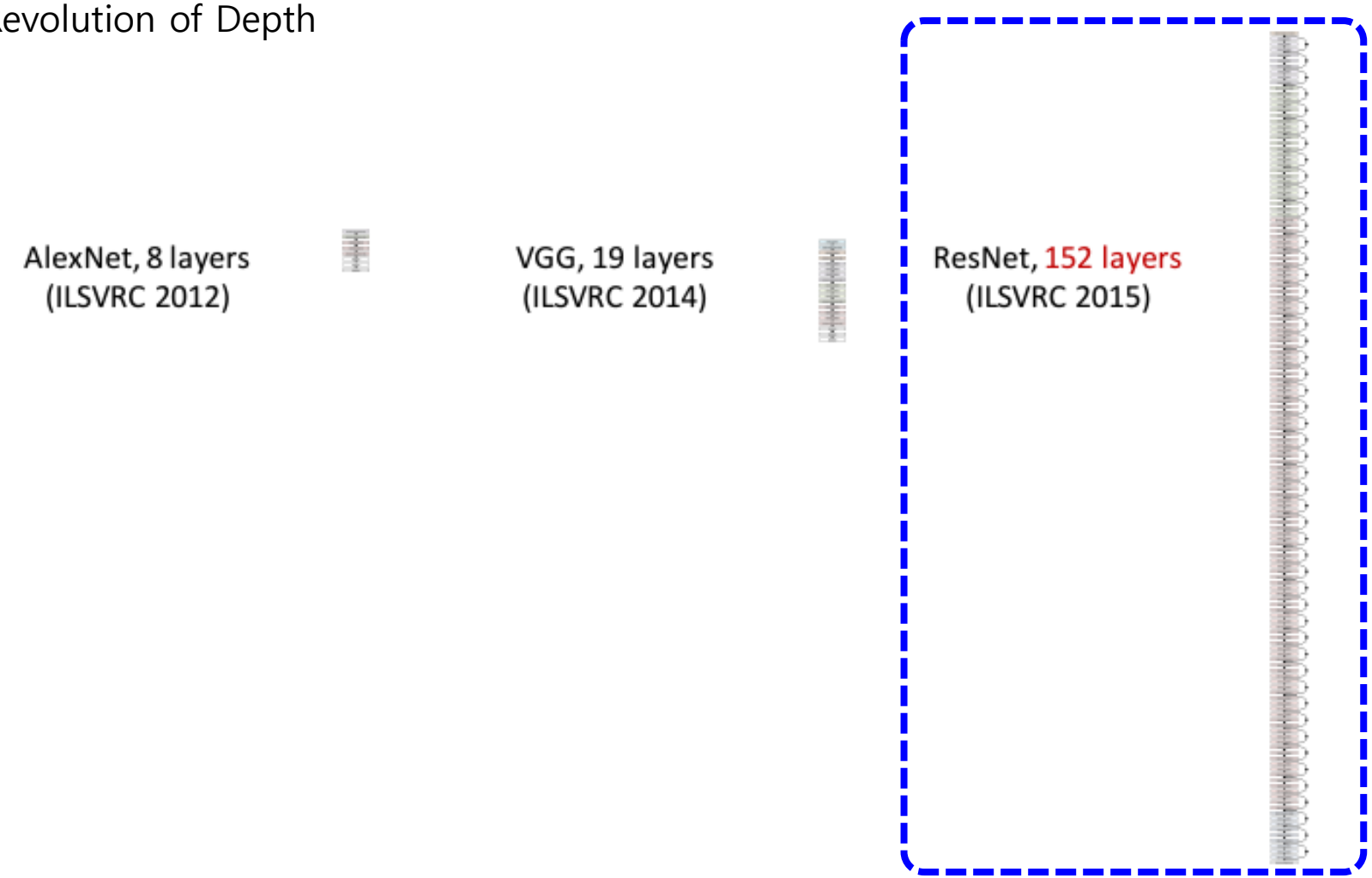
GoogleNet, 22 layers
(ILSVRC 2014)

KaimingHe,XiangyuZhang,ShaoqingRen,&JianSun."DeepResidualLearningforImageRecognition".CVPR2016

Convolutional Neural Network (CNN)

■ CNN Architecture 모델

□ Revolution of Depth



KaimingHe,XiangyuZhang,ShaoqingRen,&JianSun."DeepResidualLearningforImageRecognition".CVPR2016

Convolutional Neural Network (CNN)

■ CNN Architecture 모델 [1]

□ VGGNet

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

- 신경망의 깊이가 모델의 성능에 미치는 영향을 조사하기 위해 연구 시작
- 단순한 연산만을 가지고 모델 구성됨 (3x3 합성곱, Max-pooling, soft-max)
- **VGG16(D), VGG19(E)**: VGGNet 에서 많이 쓰이는 필터 종류

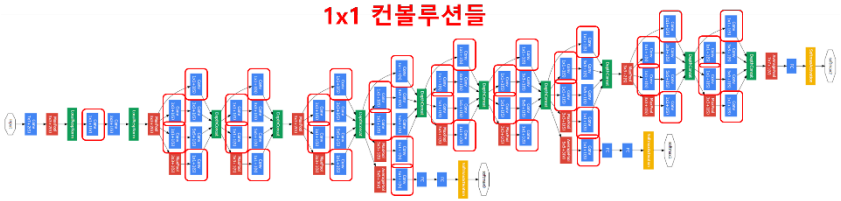
Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).-

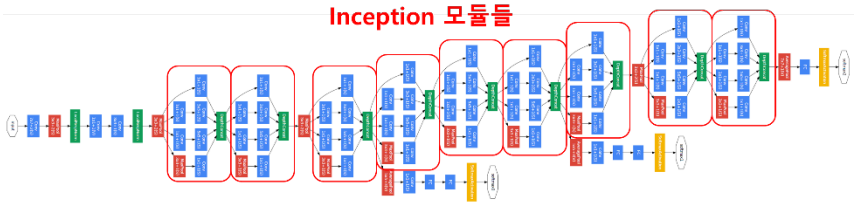
Convolutional Neural Network (CNN)

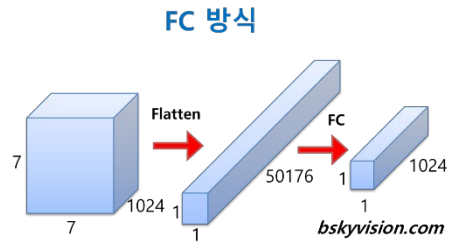
■ CNN Architecture 모델 [2]

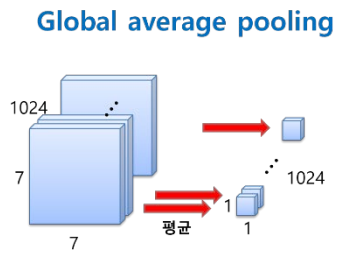
□ GoogLeNet

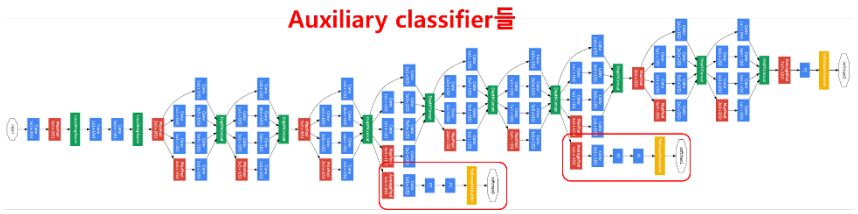
- 2014년 이미지넷 이미지 인식 대회 (ILSVRC)에서 VGGNet (VGG19)을 이기고 우승을 차지한 알고리즘
- 19층의 VGG19보다 좀 더 깊은 22층으로 구성 (파란색 블럭의 층수를 세보면 22개 층)
- Inception module (인셉션 모듈)이라는 블록을 가지고 있음









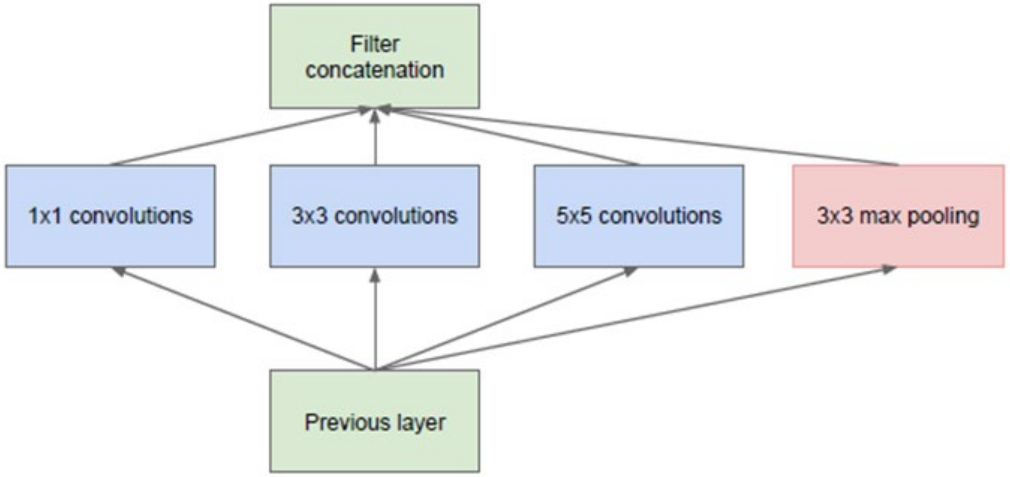


Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

Convolutional Neural Network (CNN)

■ CNN Architecture 모델 [2]

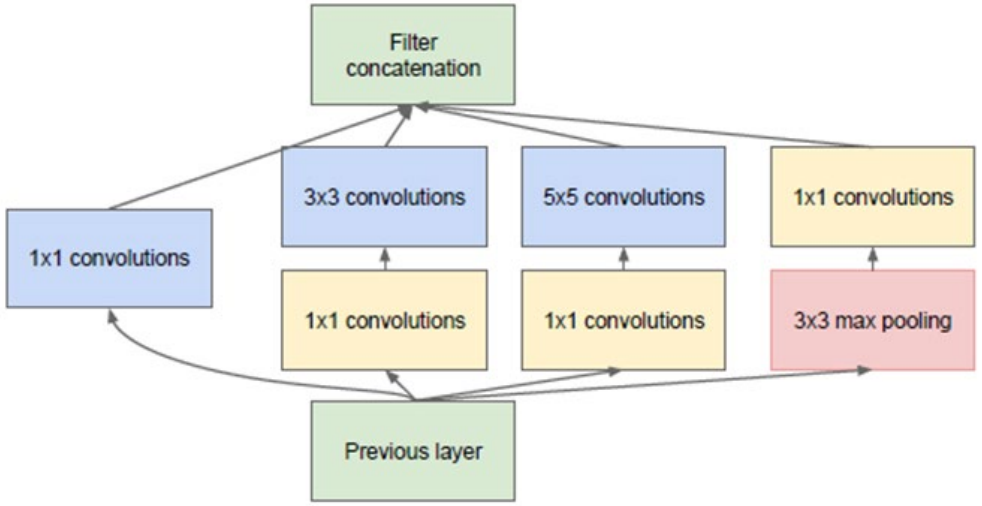
□ GoogLeNet



- 기초적인 인셉션 모듈
- 인셉션 모듈은 이전 단계의 활성화 지도에 다양한 필터 크기 (1x1, 3x3, 5x5)로 합성곱 연산을 적용함

- **차원 감소를 더한 인셉션 모듈**
- 인셉션 모듈은 이전 단계의 활성화 지도에 1x1 합성곱이 먼저 추가됨
  -> **많은 메모리 사용을 줄이기 위함**

- 이후 다양한 필터 크기 (1x1, 3x3, 5x5)로 합성곱 연산을 적용함



Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

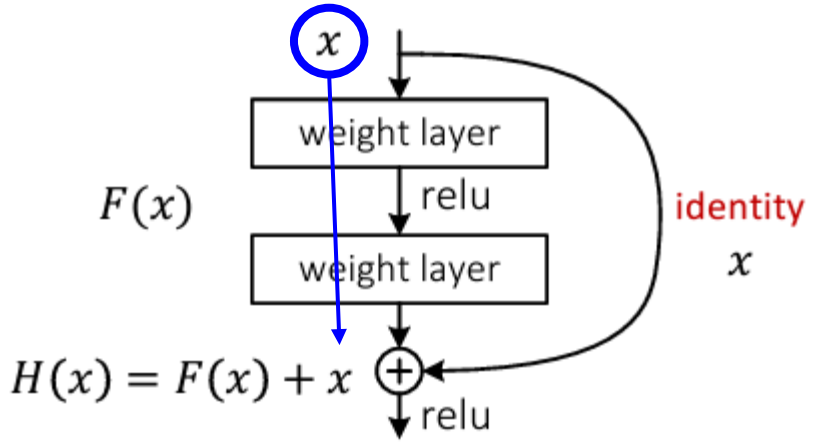
Convolutional Neural Network (CNN)

■ CNN Architecture 모델 [3]

□ ResNet

- 일정 수준 이상의 깊이가 되면 오히려 얕은 모델보다 깊은 모델의 성능이 떨어진다는 점을 발견
- 이 문제 해결을 위해 잔차 학습 (Residual learning) 제시

특정 위치에서 입력이 들어왔을 때
합성곱 연산을 통과한 결과와 **입력으로 들어온 결과** 두 가지를 더하여 **다음 레이어에 전달**



- **잔차 학습 블록**
  - 이전 단계에서 뽑았던 특성을 변형시키지 않고 그대로 더해서 전달
  - 입력 단에 가까운 곳에서 뽑은 단순한 특성과 뒷부분에서 뽑은 복잡한 특성 모두를 사용하는 장점
  - 더하기 연산은 역전파 계산 시 기울기가 1이기 때문에 손실의 변화 없이 앞부분까지 전파가 잘됨

Kaiming, He. "Deep residual networks: deep learning gets way deeper." Proceedings of the ICML 2015 32nd International Conference on Machine Learning. 2015.

Thank you

KOREA
UNIVERSITY