Medical Image Analysis

3. Medical image classification(2)

Taeyang Yang

Oct. 2020

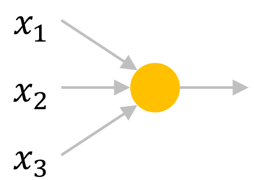
https://www.edwith.org/medical-20200327/joinLectures/30437

Contents

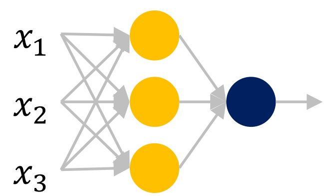
- Deep neural network 구조
- Convolutional neural network 구조
- 주요 네트워크 구조
- 네트워크를 의료영상 분류에 적용하기 위한 방법

1. Property of Deep Neural Network

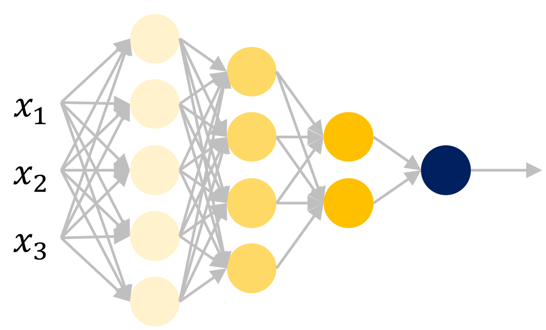
- Logistic regression



- Neural network

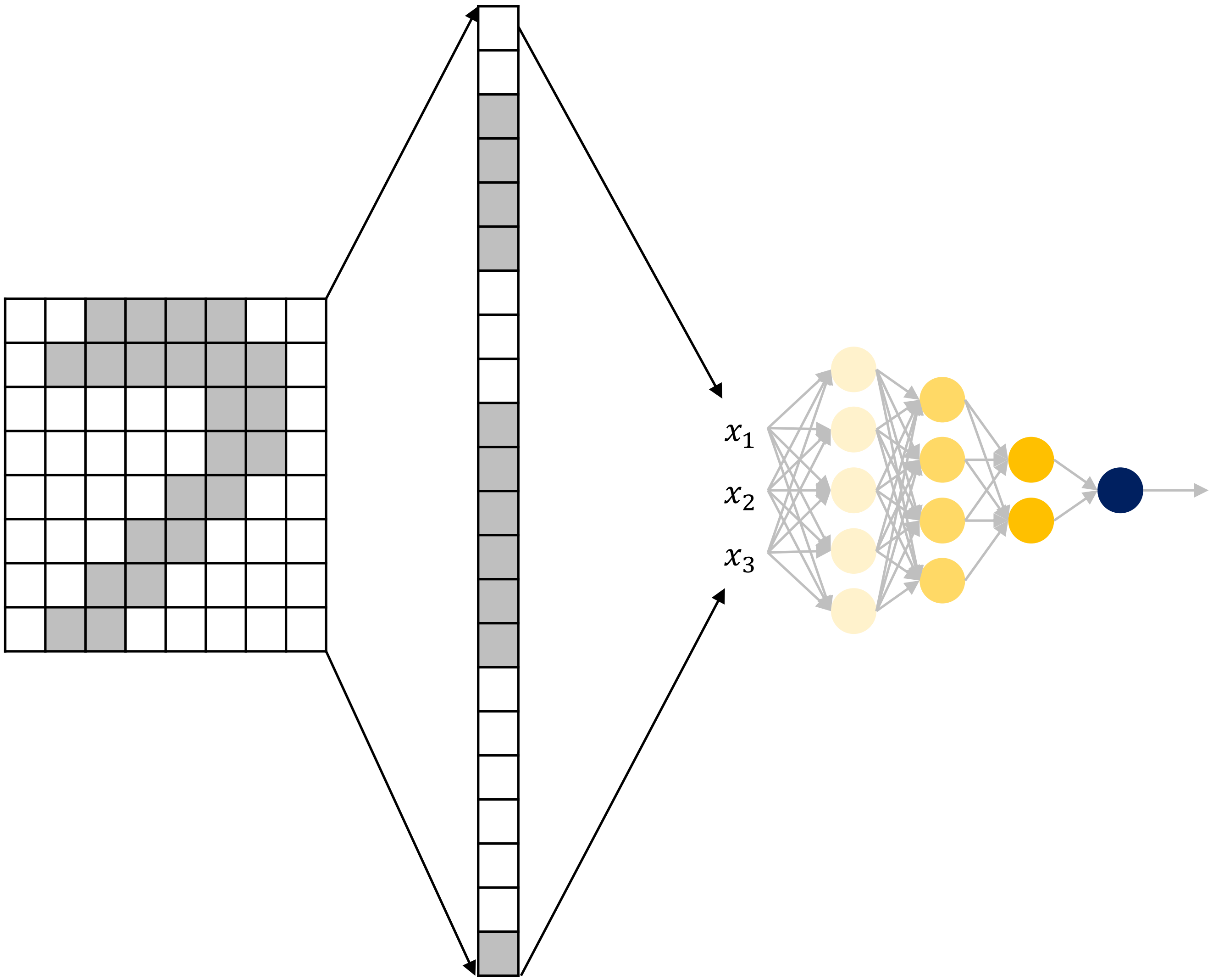


- Deep neural network



1. Property of Deep Neural Network

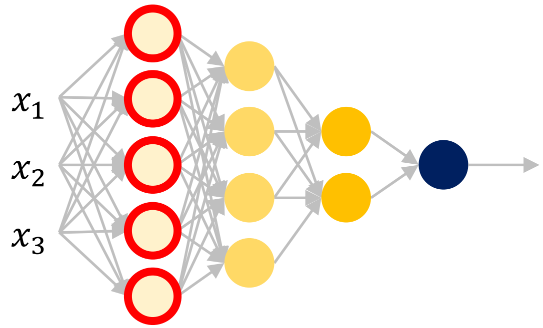
Image input

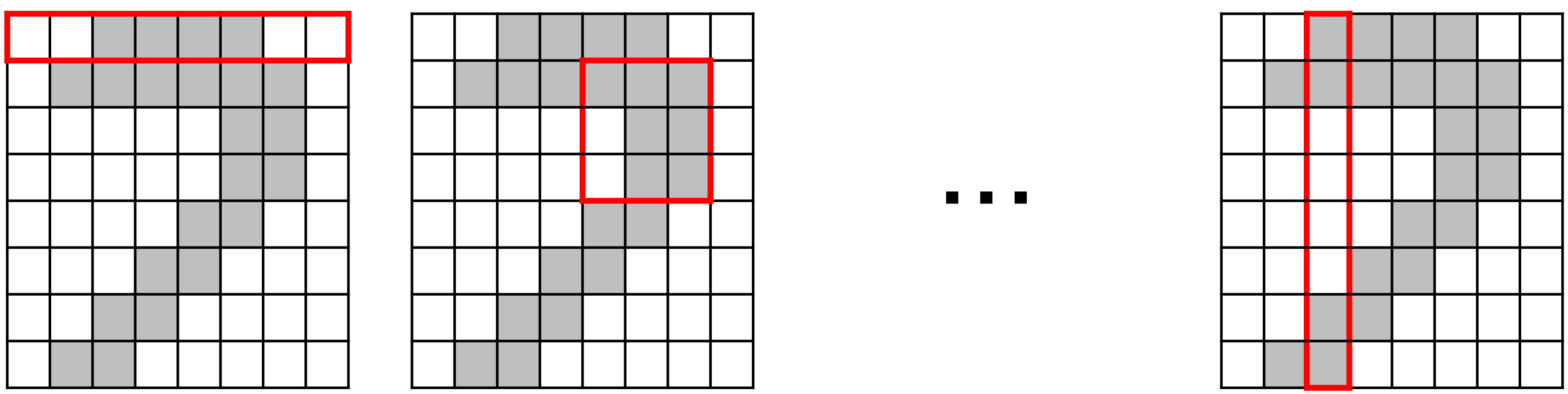


1. Property of Deep Neural Network

Role of Deep hidden layers

- 각각의 node들은 서로 다른 영역을 맡아 체크한다.
- Layer마다 체크하는 수준이 달라진다.
  - 1st layer: low level



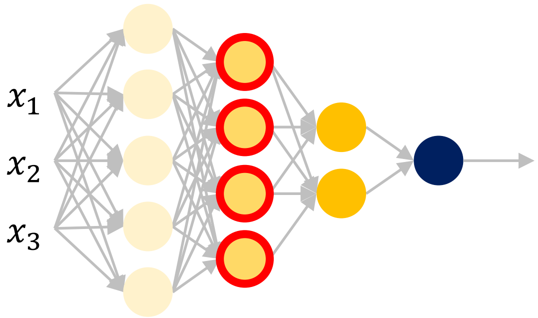


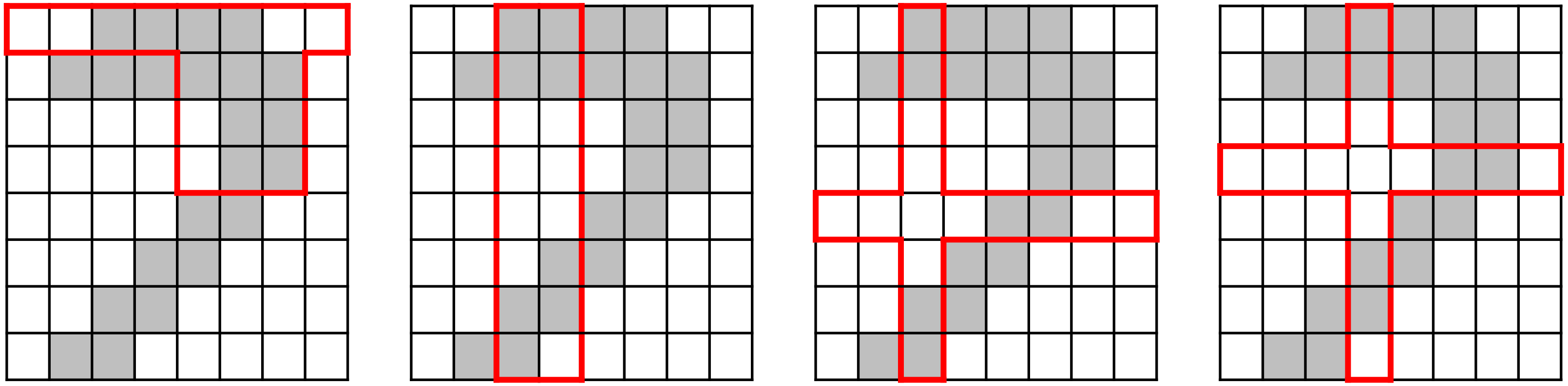
...

1. Property of Deep Neural Network

Role of Deep hidden layers

- 각각의 node들은 서로 다른 영역을 맡아 체크한다.
- Layer마다 체크하는 수준이 달라진다.
    - 2nd layer: high level

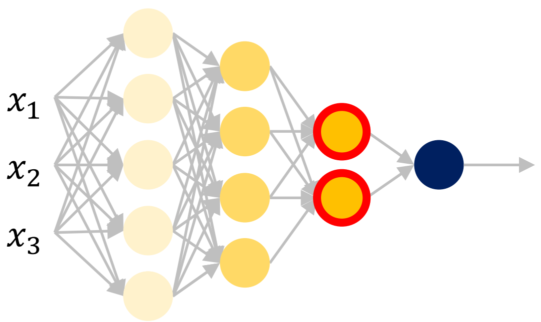


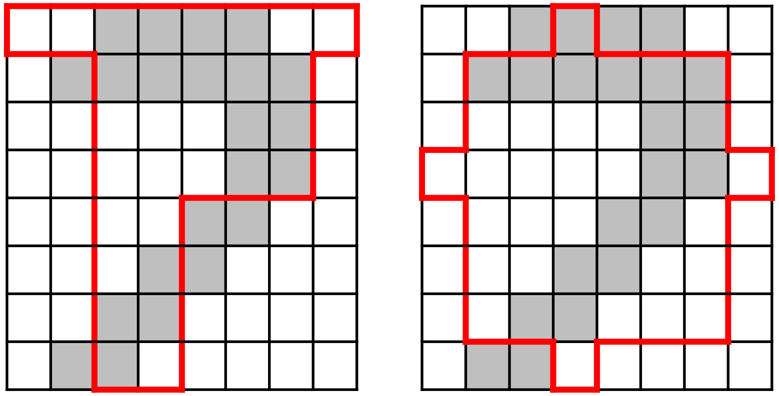


1. Property of Deep Neural Network

Role of Deep hidden layers

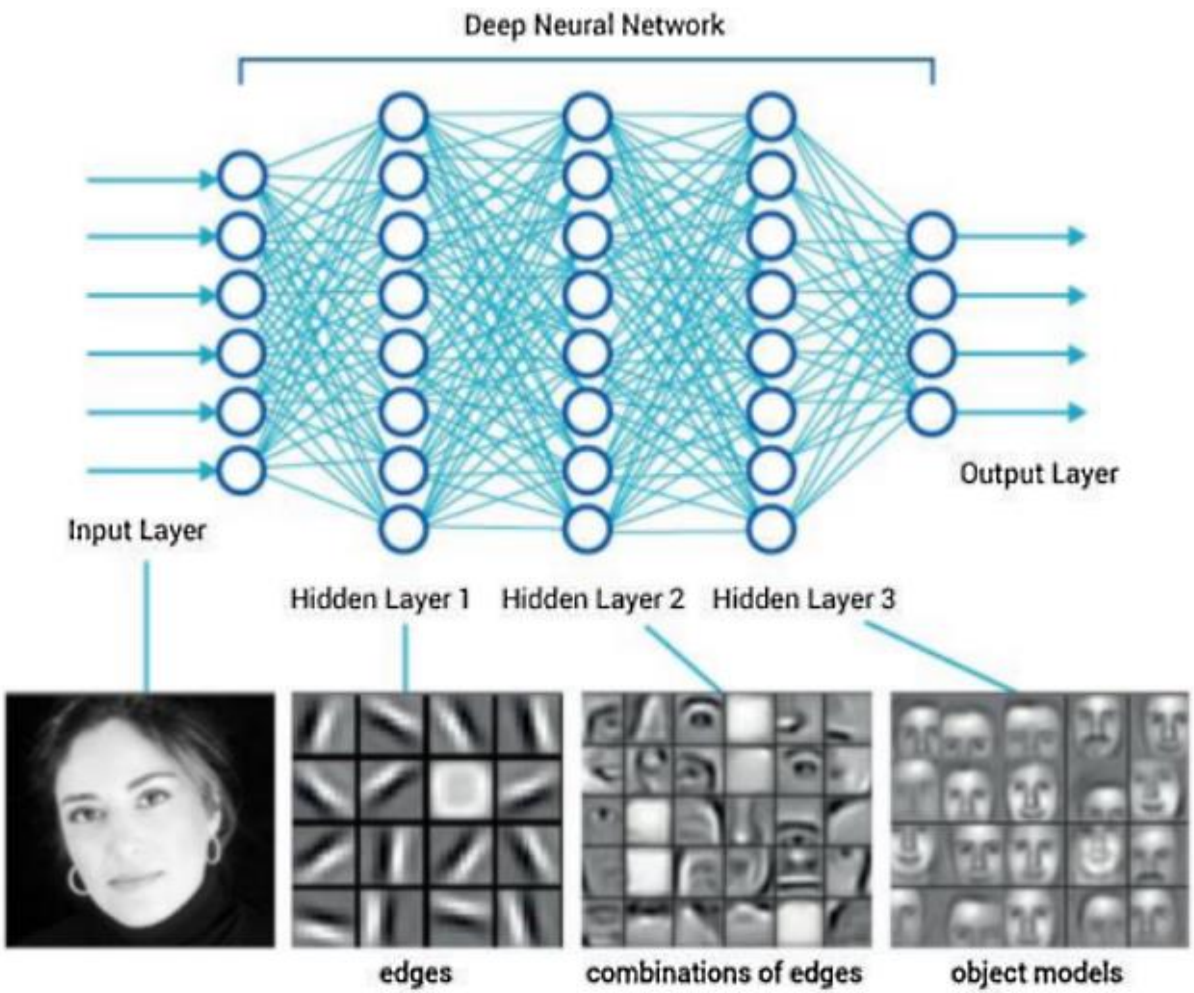
- 각각의 node들은 서로 다른 영역을 맡아 체크한다.
- Layer마다 체크하는 수준이 달라진다.
  - 3rd layer: higher level





1. Property of Deep Neural Network

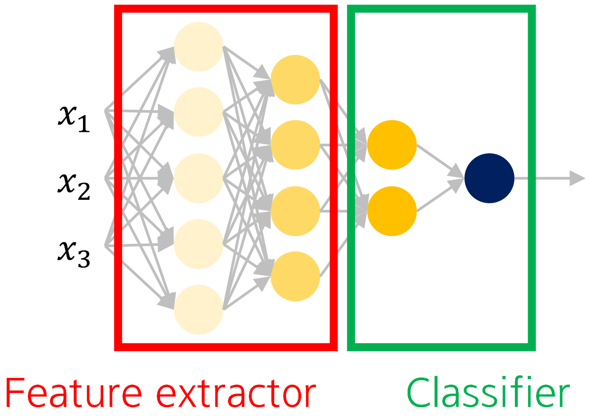
Role of Deep hidden layers



https://iq.opengenus.org/hidden-layers/

1. Property of Deep Neural Network

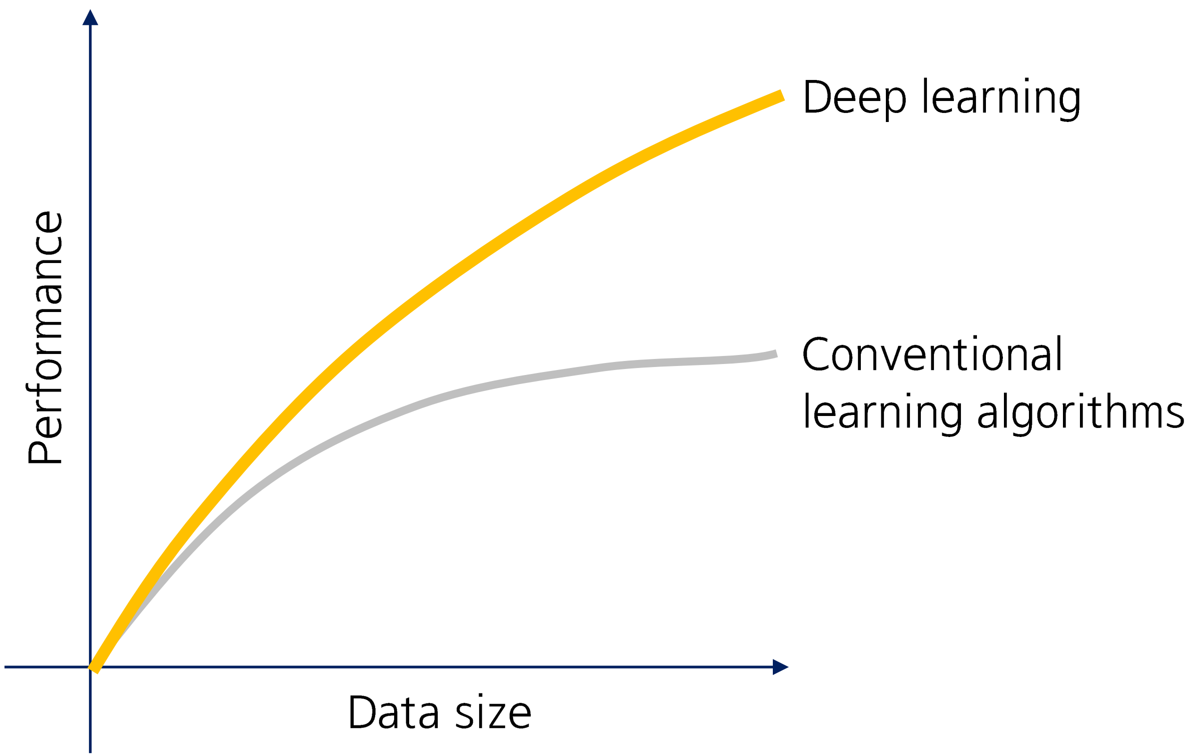
Deep neural network: End-to-end learning (Feature extractor + Classifier)



1. Property of Deep Neural Network

Important of data size

- Data size에 따라 점차 성능이 좋아지는 Deep neural network
  - Data size가 적은 의료영상 도메인 특성상 Conventional learning algorithm이 나은 경우도 많다.



1. Property of Deep Neural Network

Limitation of deep neural network

- Too many parameters

2. Convolution

Convolution filter

Data

9	9	9	0	0	0
9	9	9	0	0	0
9	9	9	0	0	0
9	9	9	0	0	0
9	9	9	0	0	0
9	9	9	0	0	0

Convolution filter

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

2. Convolution

Convolution

Data

9 1/9	9 1/9	9 1/9	0	0	0
9 1/9	9 1/9	9 1/9	0	0	0
9 1/9	9 1/9	9 1/9	0	0	0
9	9	9	0	0	0
9	9	9	0	0	0
9	9	9	0	0	0

Output

9			

$$9 \times \frac{1}{9} + 9 \times \frac{1}{9} + 9 \times \frac{1}{9} + 9 \times \frac{1}{9} + 9 \times \frac{1}{9} + 9 \times \frac{1}{9} + 9 \times \frac{1}{9} + 9 \times \frac{1}{9} + 9 \times \frac{1}{9} = 9$$

Convolution

Data

9	9 1/9	9 1/9	0 1/9	0	0
9	9 1/9	9 1/9	0 1/9	0	0
9	9 1/9	9 1/9	0 1/9	0	0
9	9	9	0	0	0
9	9	9	0	0	0
9	9	9	0	0	0

Output

9	6		

$$9 \times \frac{1}{9} + 9 \times \frac{1}{9} + 9 \times \frac{1}{9} + 9 \times \frac{1}{9} + 9 \times \frac{1}{9} + 9 \times \frac{1}{9} + 0 \times \frac{1}{9} + 0 \times \frac{1}{9} + 0 \times \frac{1}{9} = 6$$

2. Convolution

Convolution

Data

9	9	9 1/9	0 1/9	0 1/9	0
9	9	9 1/9	0 1/9	0 1/9	0
9	9	9 1/9	0 1/9	0 1/9	0
9	9	9	0	0	0
9	9	9	0	0	0
9	9	9	0	0	0

Output

9	6	3	

$$9 \times \frac{1}{9} + 9 \times \frac{1}{9} + 9 \times \frac{1}{9} + 0 \times \frac{1}{9} + 0 \times \frac{1}{9} + 0 \times \frac{1}{9} + 0 \times \frac{1}{9} + 0 \times \frac{1}{9} + 0 \times \frac{1}{9} = 6$$

2. Convolution

Convolution

Data

9	9	9	0	0	0
9	9	9	0	0	0
9	9	9	0	0	0
9	9	9	0 1/9	0 1/9	0 1/9
9	9	9	0 1/9	0 1/9	0 1/9
9	9	9	0 1/9	0 1/9	0 1/9

Output

9	6	3	0
9	6	3	0
9	6	3	0
9	6	3	0

$$9 \times \frac{1}{9} + 9 \times \frac{1}{9} + 9 \times \frac{1}{9} + 0 \times \frac{1}{9} + 0 \times \frac{1}{9} + 0 \times \frac{1}{9} + 0 \times \frac{1}{9} + 0 \times \frac{1}{9} + 0 \times \frac{1}{9} = 6$$

2. Convolution

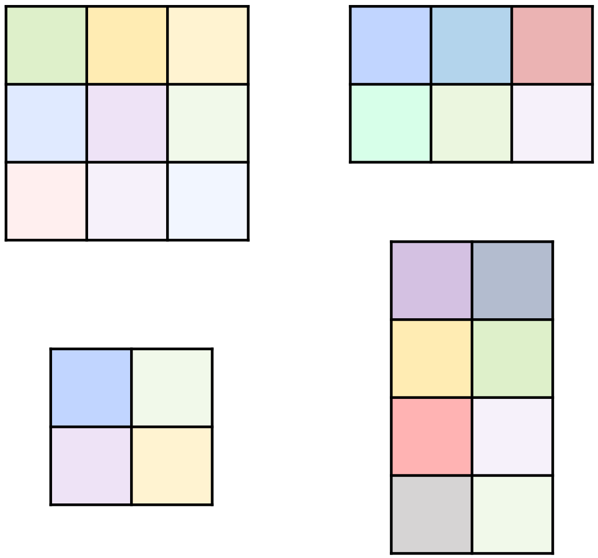
Different Convolution filter

1/9	1/9	1/9
1/9	-17/9	1/9
1/9	1/9	1/9

2. Convolution

Diverse Convolution filter

- 다양한 필터가 만들어질 수 있다.
- 단, 같은 레이어에서는 같은 크기의 필터가 사용된다.



2. Convolution

Convolution examples

- 각 필터는 다양한 역할을 수행한다

Input | **Convolution filter** | **Feature**

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Edge

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Blurred

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Sharpen

3. Convolutional Neural Network (CNN)

Pooling layer

Data

9 1/9	9 1/9	9 1/9	0	0	0
9 1/9	9 1/9	9 1/9	0	0	0
9 1/9	9 1/9	9 1/9	0	0	0
9	9	9	0	0	0
9	9	9	0	0	0
9	9	9	0	0	0

Convolutional layer

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Output

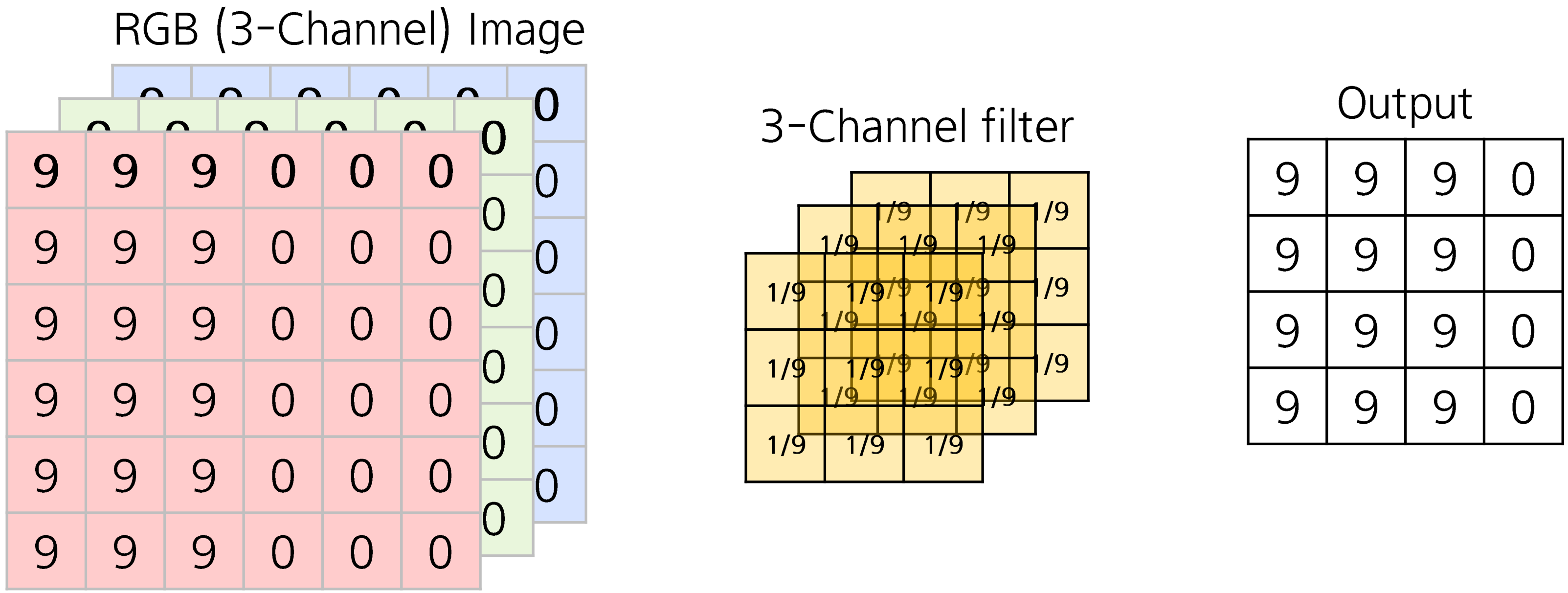
9	6	3	0
9	6	3	0
9	6	3	0
9	6	3	0

$$9 \times \frac{1}{9} + 9 \times \frac{1}{9} + 9 \times \frac{1}{9} + 9 \times \frac{1}{9} + 9 \times \frac{1}{9} + 9 \times \frac{1}{9} + 9 \times \frac{1}{9} + 9 \times \frac{1}{9} + 9 \times \frac{1}{9} = 9$$

3. Convolutional Neural Network (CNN)

RGB image convolution

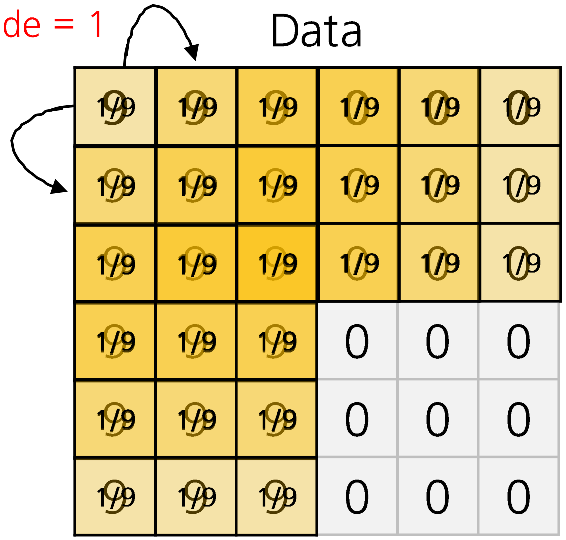
- 입력 이미지의 채널에 맞추어 Convolution filter의 크기도 조정된다



3. Convolutional Neural Network (CNN)

Strided convolution

Stride = 1



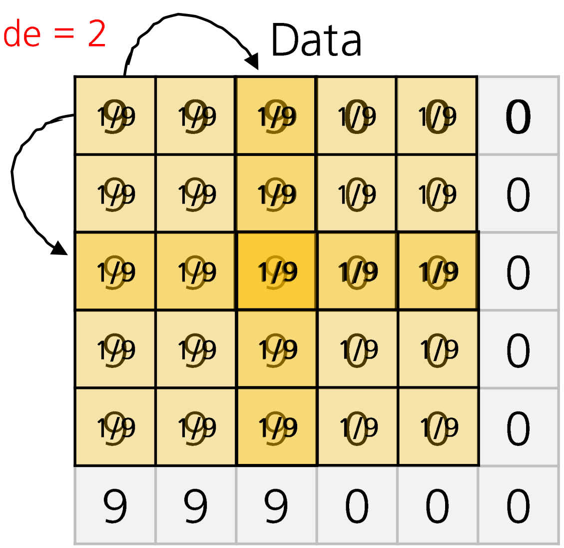
Convolutional layer

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Output

9	6	3	0
9	6	3	0
9	6	3	0
9	6	3	0

Stride = 2



Convolutional layer

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Output

9	3
9	3

3. Convolutional Neural Network (CNN)

Padding

Padding = 1

Data

0	0	0	0	0	0	0	0
1/9 0	1/9 9	1/9 9	9	0	0	0	0
1/9 0	1/9 9	1/9 9	9	0	0	0	0
1/9 0	1/9 9	1/9 9	9	0	0	0	0
0	9	9	9	0	0	0	0
0	9	9	9	0	0	0	0
0	9	9	9	0	0	0	0
0	0	0	0	0	0	0	0

Convolutional layer

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Output

4	6	4	2	0	0
6	9	6	3	0	0
6	9	6	3	0	0
6	9	6	3	0	0
6	9	6	3	0	0
4	6	4	2	0	0

3. Convolutional Neural Network (CNN)

Max Pooling layer

Data

9	9	9	0	0	0
9	9	9	0	0	0
9	9	9	0	0	0
9	9	9	0	0	0
9	9	9	0	0	0
9	9	9	0	0	0

Max Pooling layer

Output

9	9	9	0
9	9	9	0
9	9	9	0
9	9	9	0

$$\max(9,9,9,9,9,9,9,9,9) = 9$$

No parameter

3. Convolutional Neural Network (CNN)

Average Pooling layer

Data

9	9	9	0	0	0
9	9	9	0	0	0
9	9	9	0	0	0
9	9	9	0	0	0
9	9	9	0	0	0
9	9	9	0	0	0

Average Pooling layer

Output

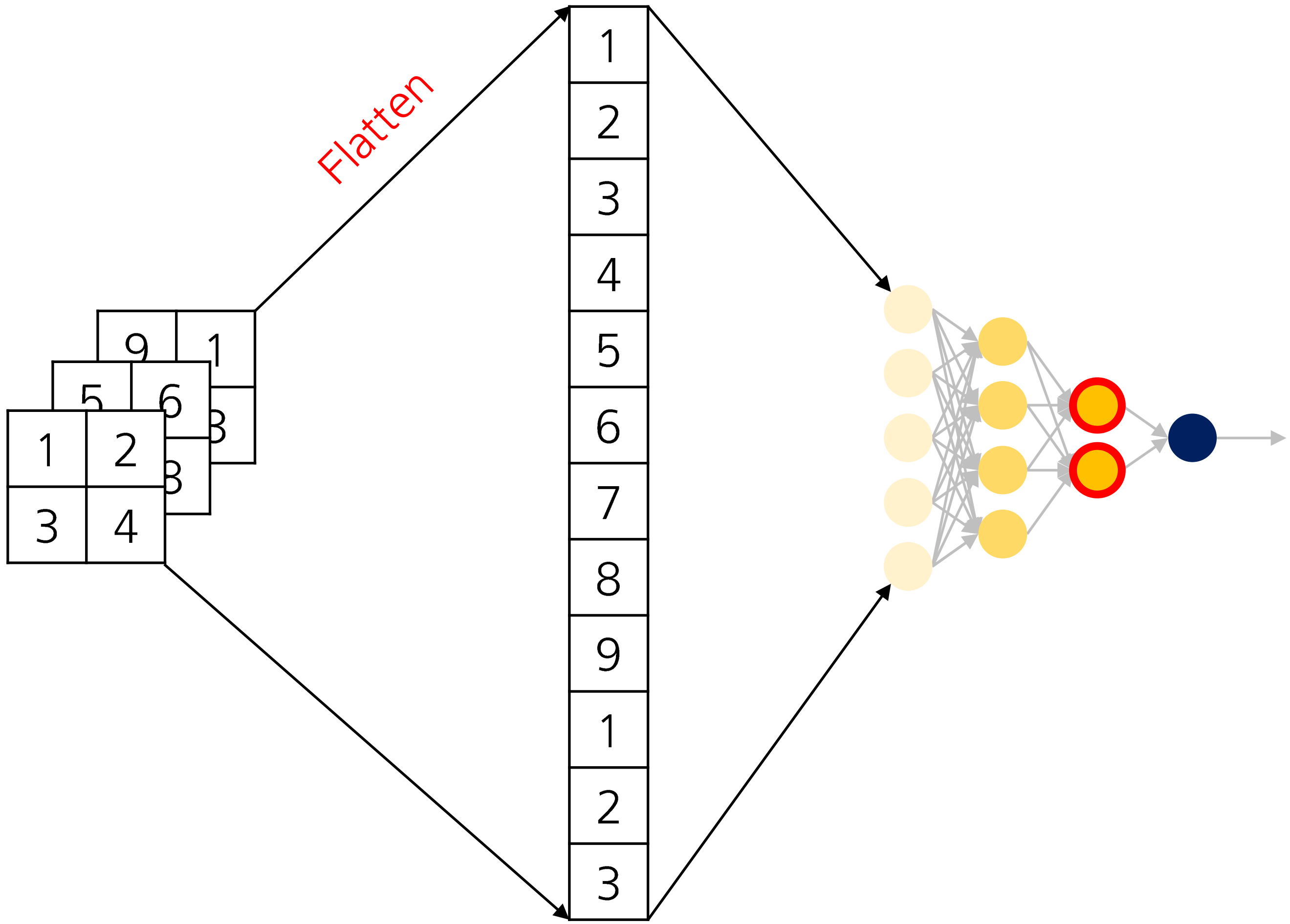
9	6	3	0
9	6	3	0
9	6	3	0
9	6	3	0

$$\frac{1}{9}(9+9+9+9+9+9+9+9+9) = 9$$

No parameter

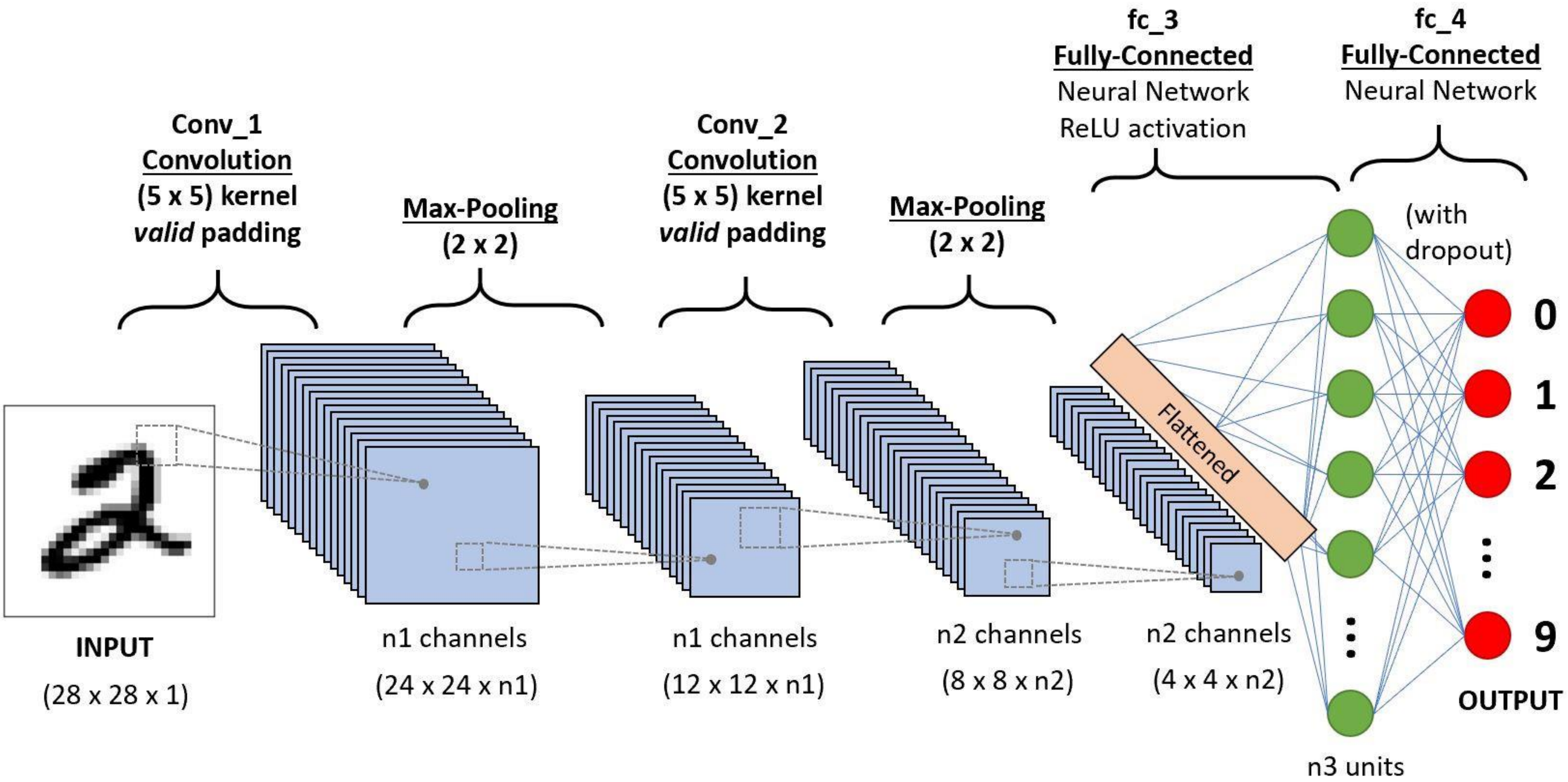
3. Convolutional Neural Network (CNN)

Fully-connected layer



3. Convolutional Neural Network (CNN)

Overall pipeline



6. 3D CNN with demographic scores

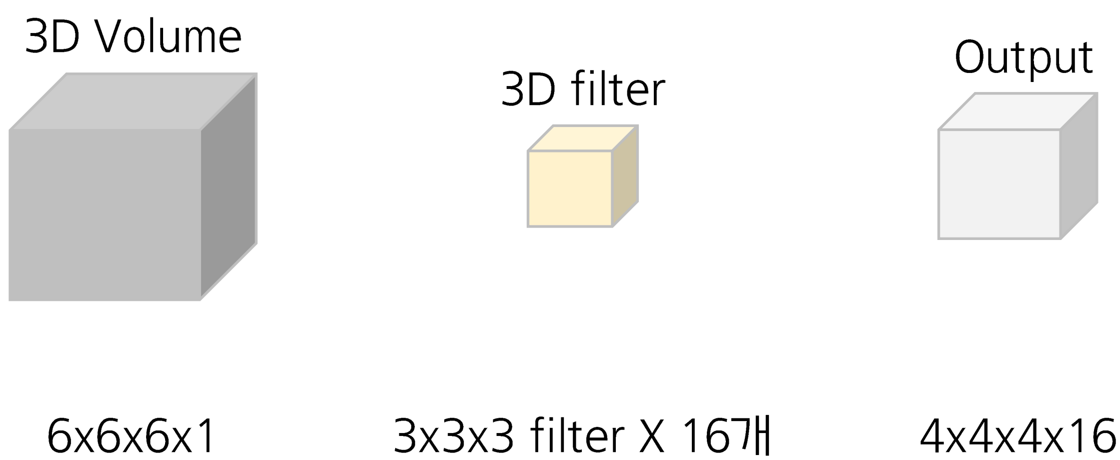
흐름상 여기가 더 어울려서 앞으로 가져옴

- 의료영상은 3D 이미지인 경우가 많음

6. 3D CNN with demographic scores

3D CNN

- 2D CNN과 거의 비슷하다



6. 3D CNN with demographic scores

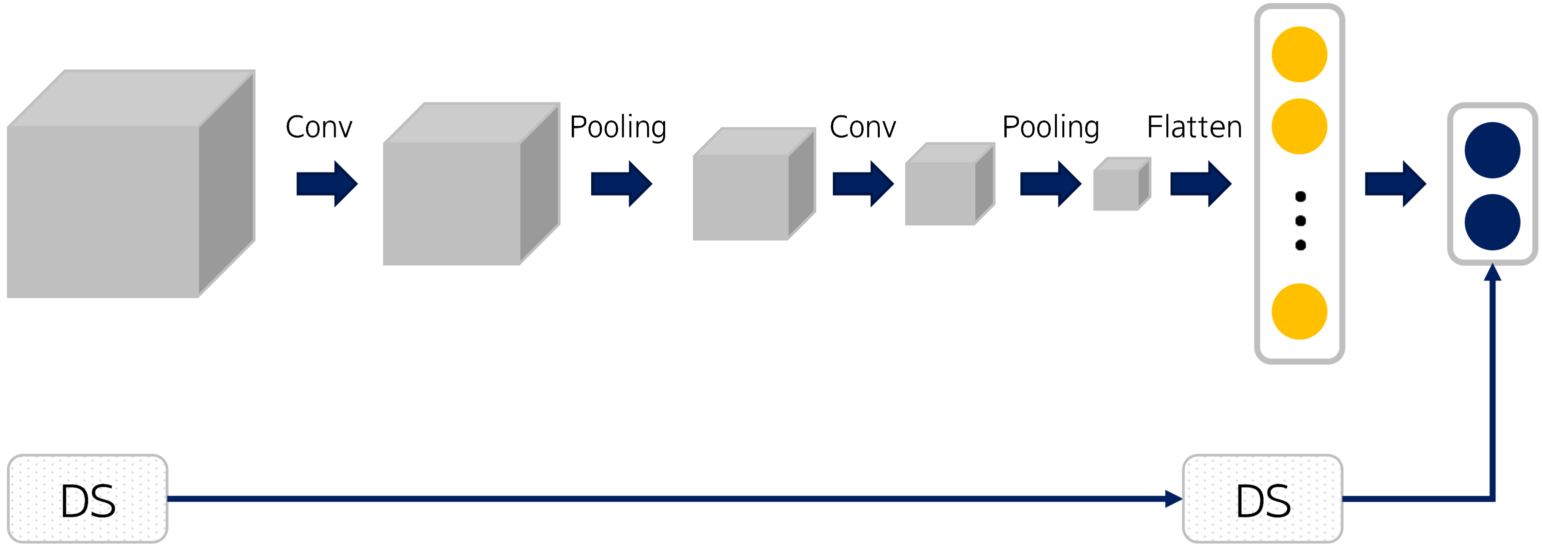
3D CNN with multiple images

- T1-weighted, T2-weighted 등 여러 이미지를 사용하는 경우
  - 인풋 채널이 증가된다고 보면 됨 (2D 예시: 흑백 이미지 → RGB 이미지)

6. 3D CNN with demographic scores

3D CNN with demographic scores

- 이런 구조가 가능함. 마찬가지로 DS의 모든 분포에 대해 충분한 샘플이 있어야 함.



4. Advanced CNNs (LeNet, AlexNet, VGG)

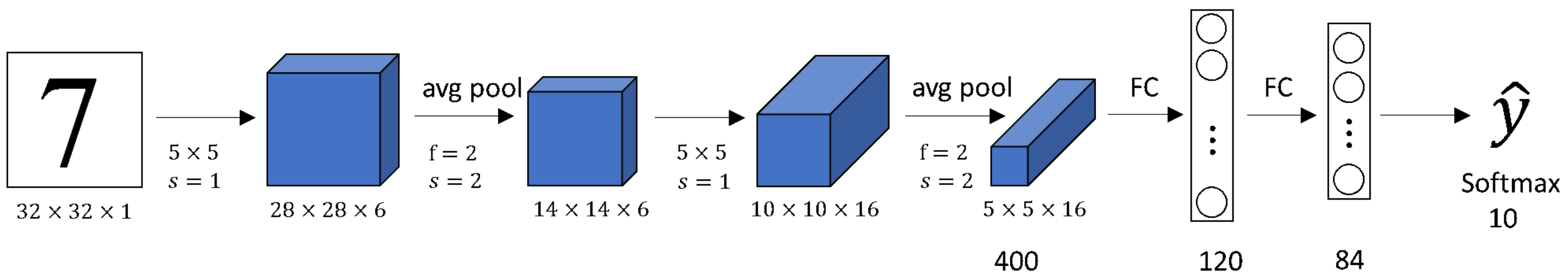
CNN architectures (1)

- LeNet-5
- AlexNet
- VGGNet
  - VGG-16
  - VGG-19

4. Advanced CNNs (LeNet, AlexNet, VGG)

LeNet-5 (1998)

뒤로 갈 수록, Feature map의 사이즈가 줄어드는 대신, 채널 수 증가

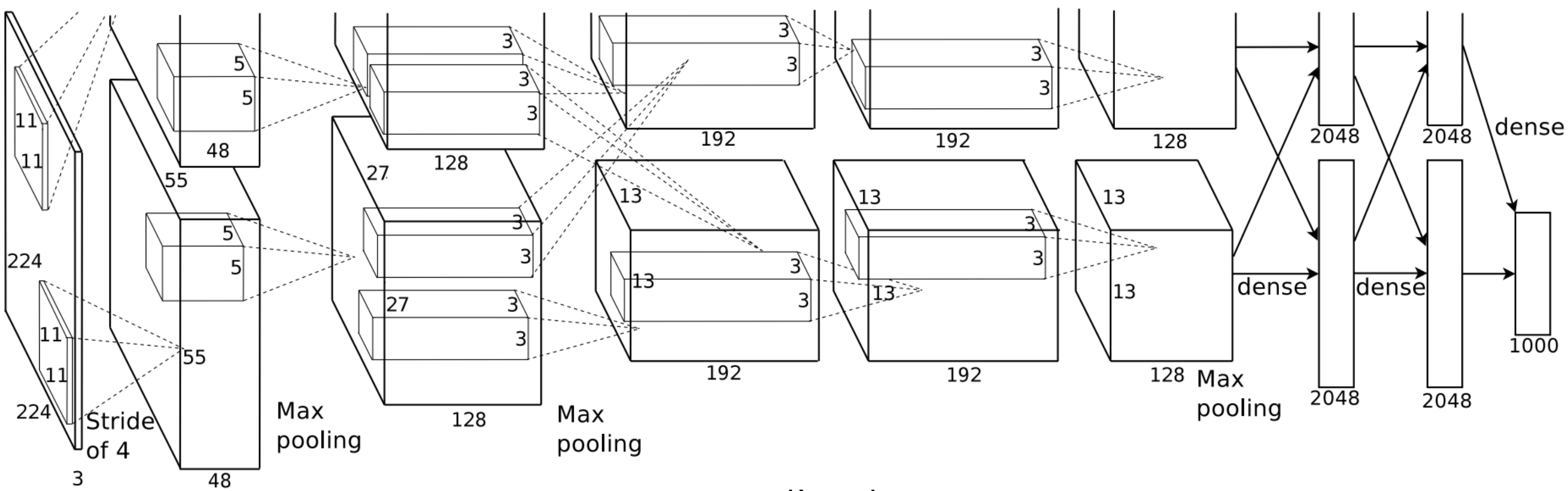


60k parameters

Simple CNN architecture

AlexNet (2012)

초반: 큰 stride로 사이즈를 크게 줄임

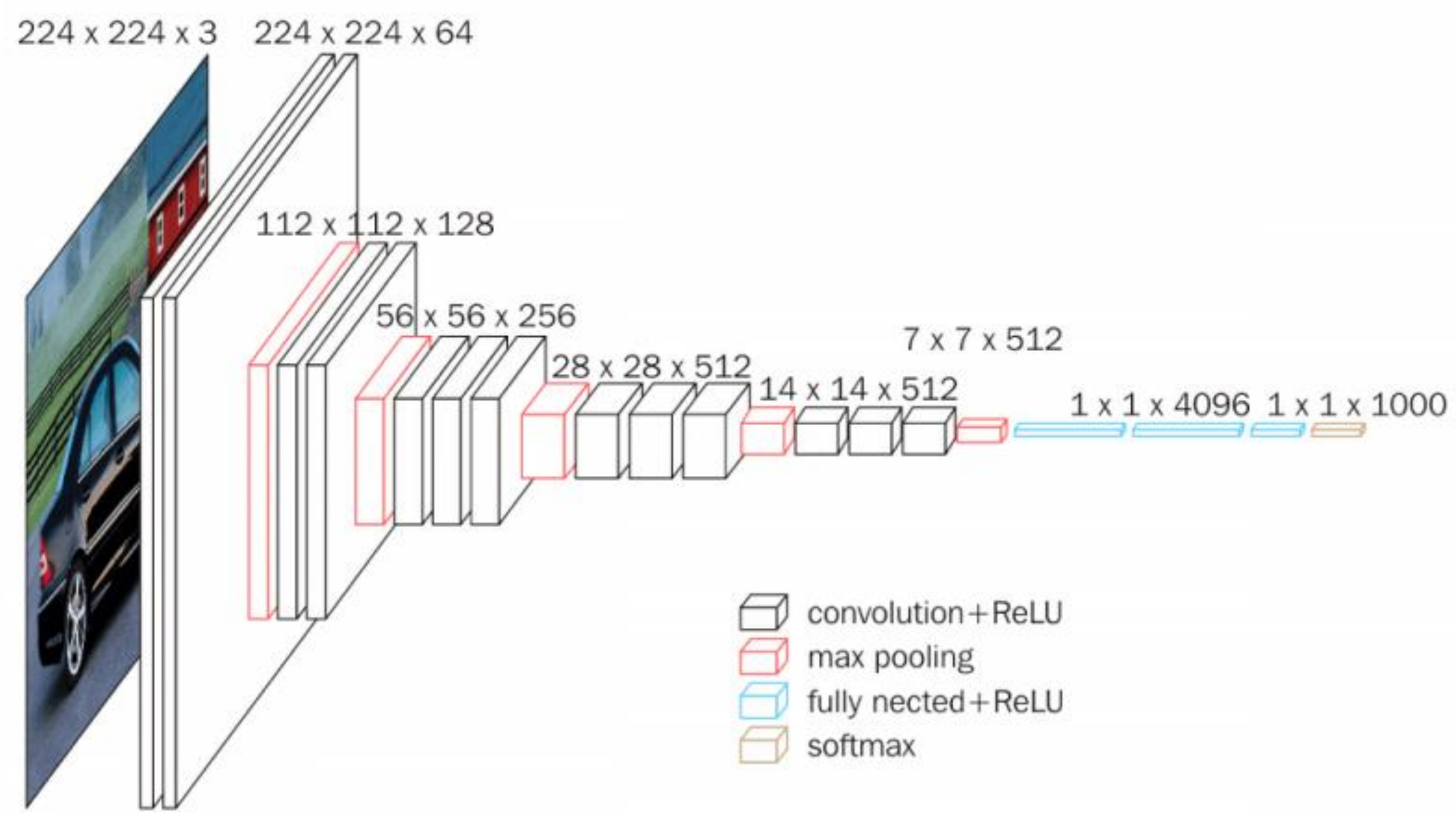


Max-pooling layer

ReLU activation function

4. Advanced CNNs (LeNet, AlexNet, VGG)

VGGNet (2015)



3x3 Convolution filter만 사용

3x3 두 번 = 5x5 필터 효과지만, 파라미터 수가 감소함

$(5 \times 5 \times 3 = 75 \rightarrow 3 \times 3 \times 3 = 27)$

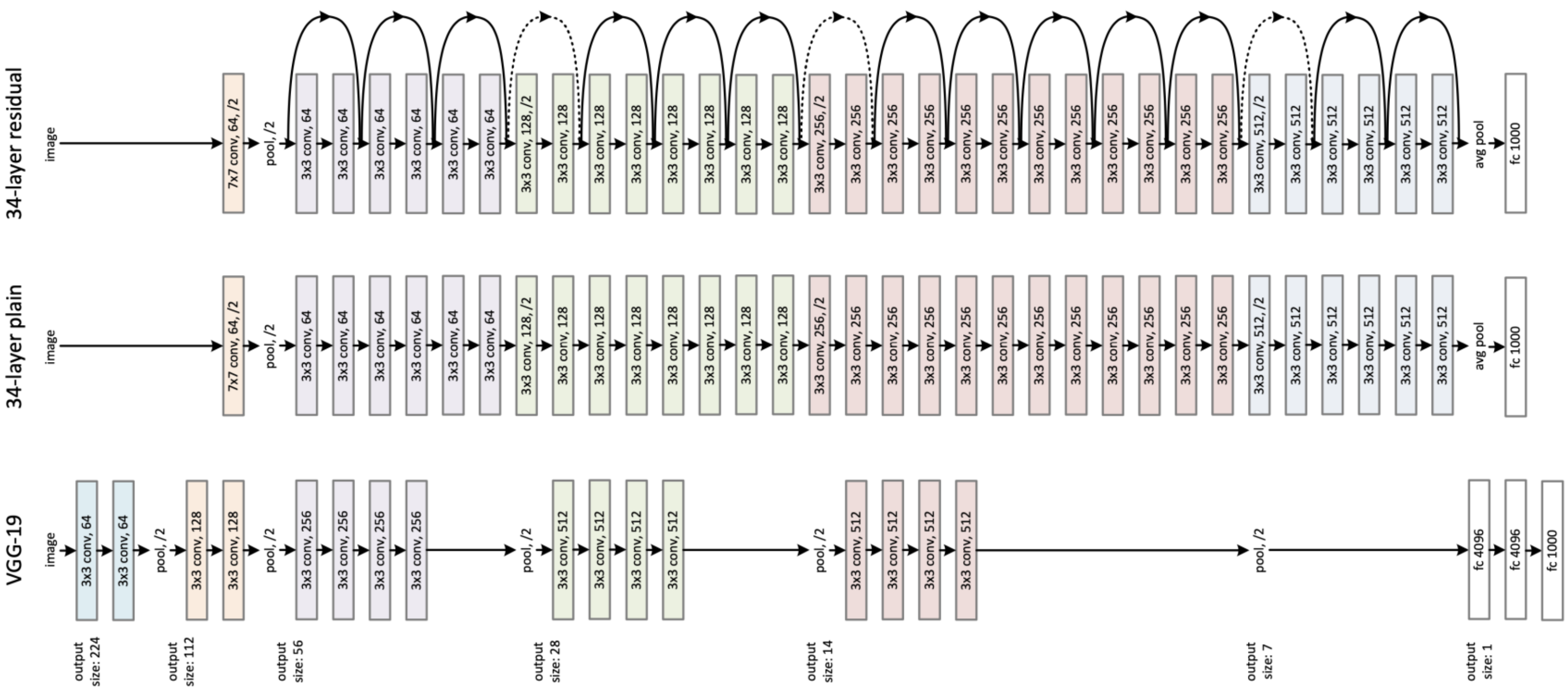
5. Advanced CNNs (ResNet, InceptionNet, DenseNet)

CNN architectures (2)

- ResNet
- Inception
- DenseNet

5. Advanced CNNs (ResNet, InceptionNet, DenseNet)

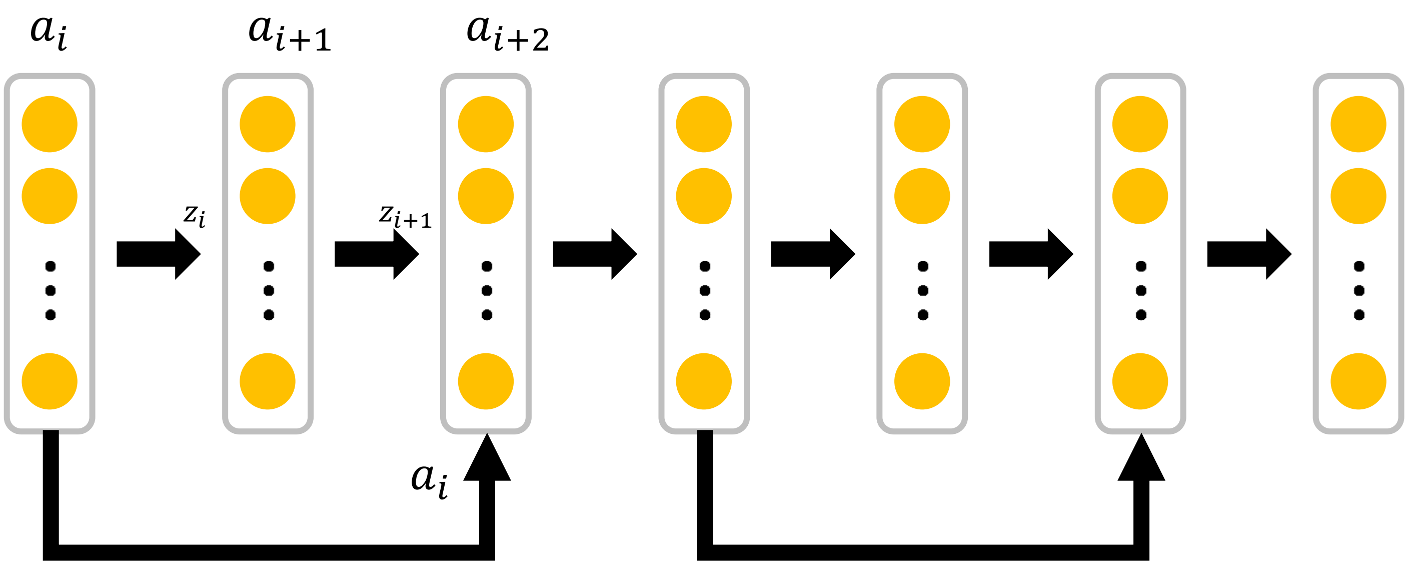
ResNet (2015)



5. Advanced CNNs (ResNet, InceptionNet, DenseNet)

ResNet (2015)

- 일반적으로 Test 할 때는, 레이어가 많으면 Overfitting 많이 됨.
- 그런데 일반적인 CNN 모델의 경우, 레이어가 너무 많으면 Training data에 대해서도 Error가 떨어짐.
- Residual block을 쓰면 이런 문제가 없음



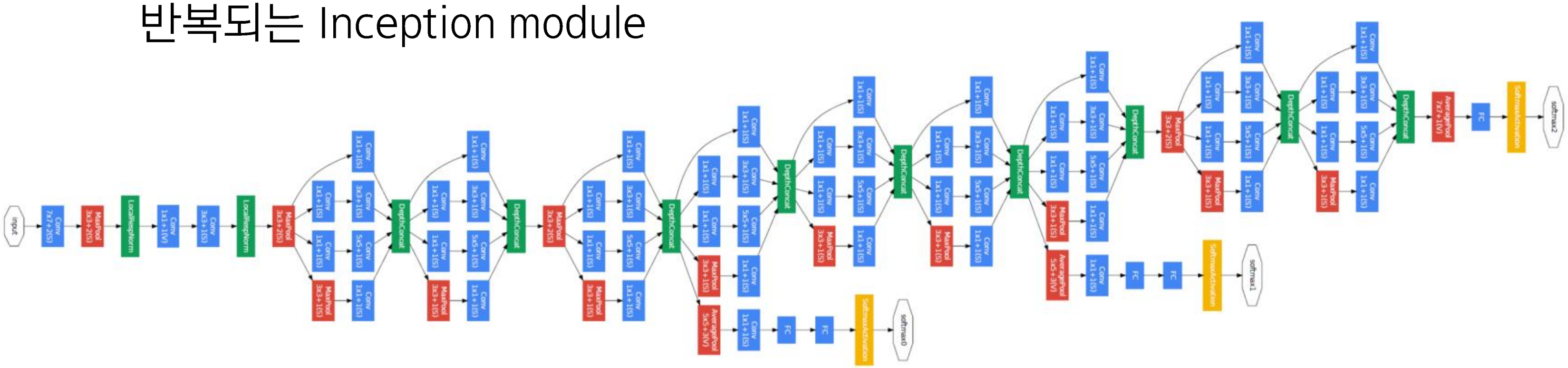
Skip connection

$$a_{i+2} = \sigma(z_{i+1} + a_i)$$

5. Advanced CNNs (ResNet, InceptionNet, DenseNet)

Inception (2014)

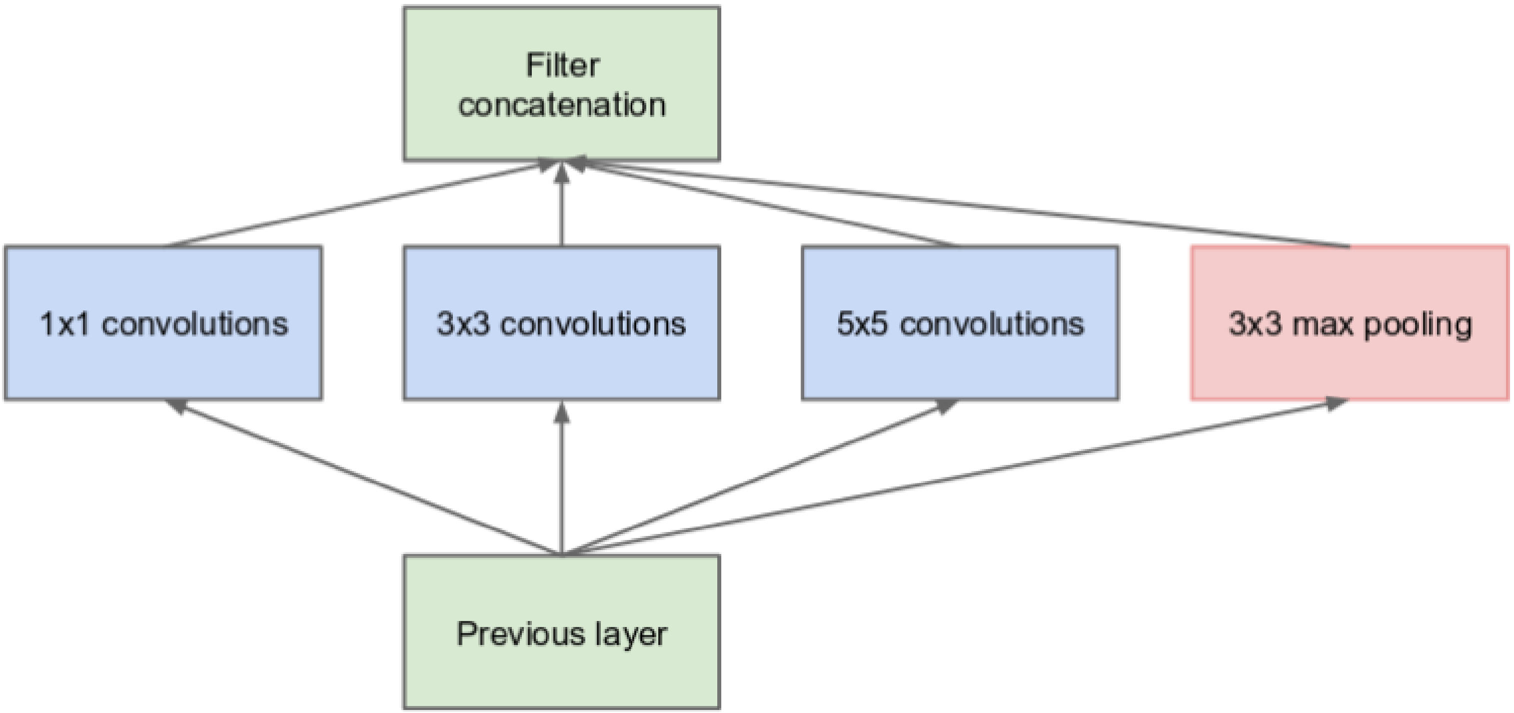
반복되는 Inception module



중간 중간 Cost 계산

5. Advanced CNNs (ResNet, InceptionNet, DenseNet)

Inception (2014) – Inception module



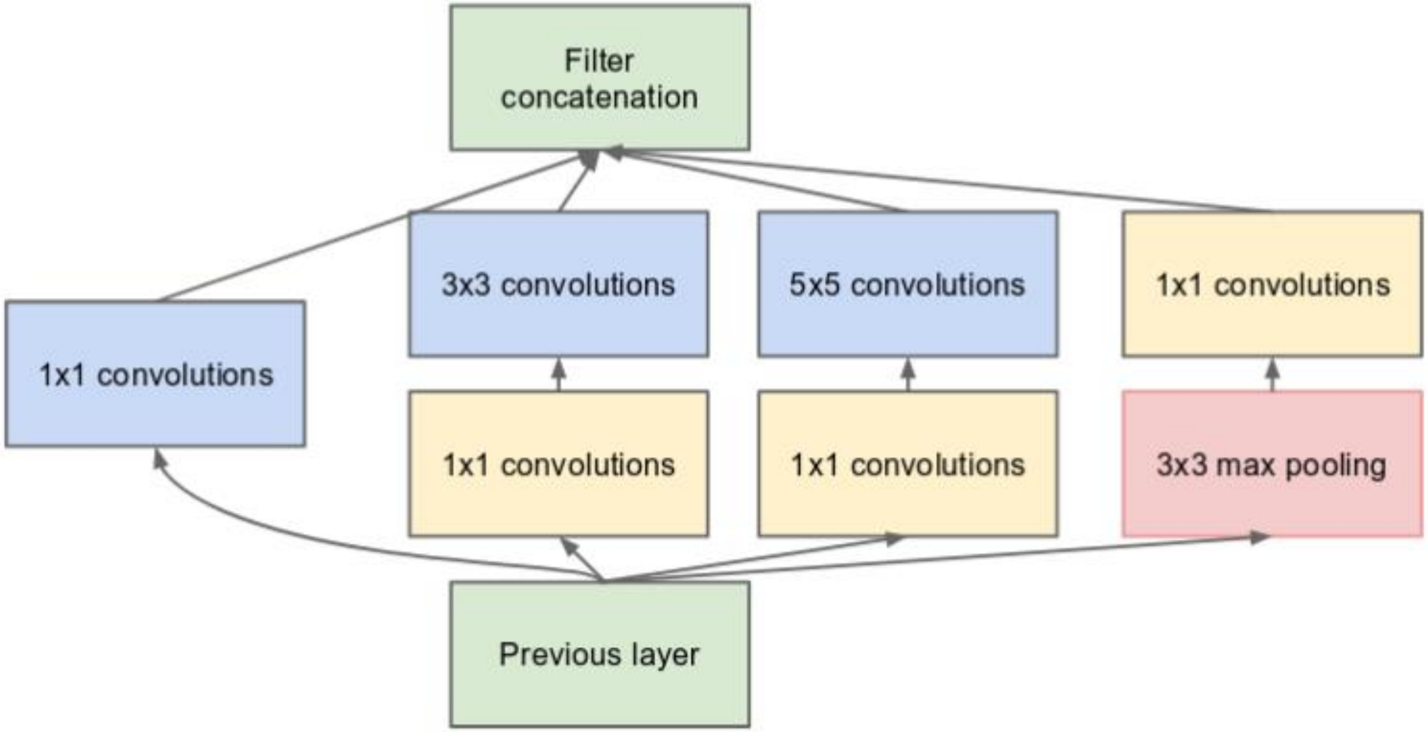
(a) Inception module, naïve version

다양한 크기의 필터를 만들어 이를 concatenate해서 사용

후반부 레이어에서는 depth (채널 개수)가 너무 깊어져서 computational cost가 너무 증가

5. Advanced CNNs (ResNet, InceptionNet, DenseNet)

Inception (2014) – Inception module with dimension reductions



(b) Inception module with dimension reductions

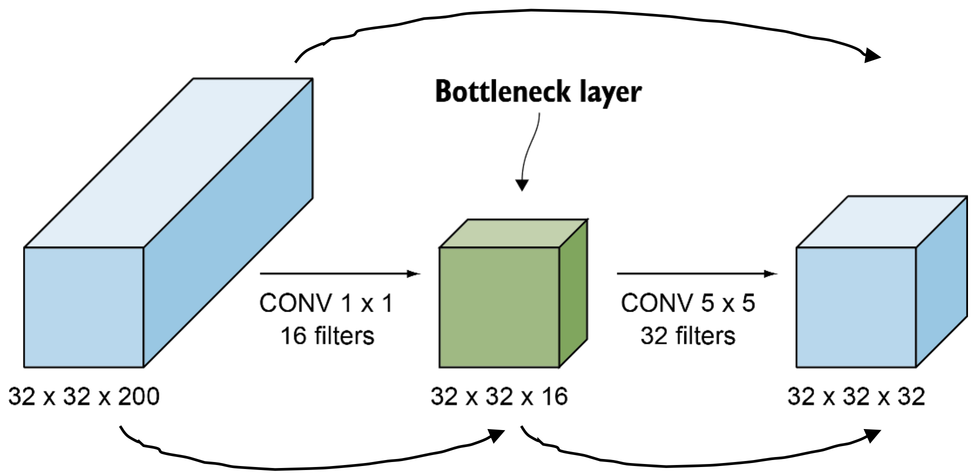
1x1 필터의 사용

5. Advanced CNNs (ResNet, InceptionNet, DenseNet)

Inception (2014) – 1x1 Conv filter

- 1x1 Conv filter
  - Depth 감소 (1x1보다 큰 필터를 써도 되긴 하지만, computational cost 최소를 위해 1x1 사용)
- 비선형성 증가
  - More activation functions
- Computational cost 감소
  - 163M ➔ 4.3M

$$(5 \times 5 \times 200) \times (32 \times 32 \times 32) = 163\text{M}$$



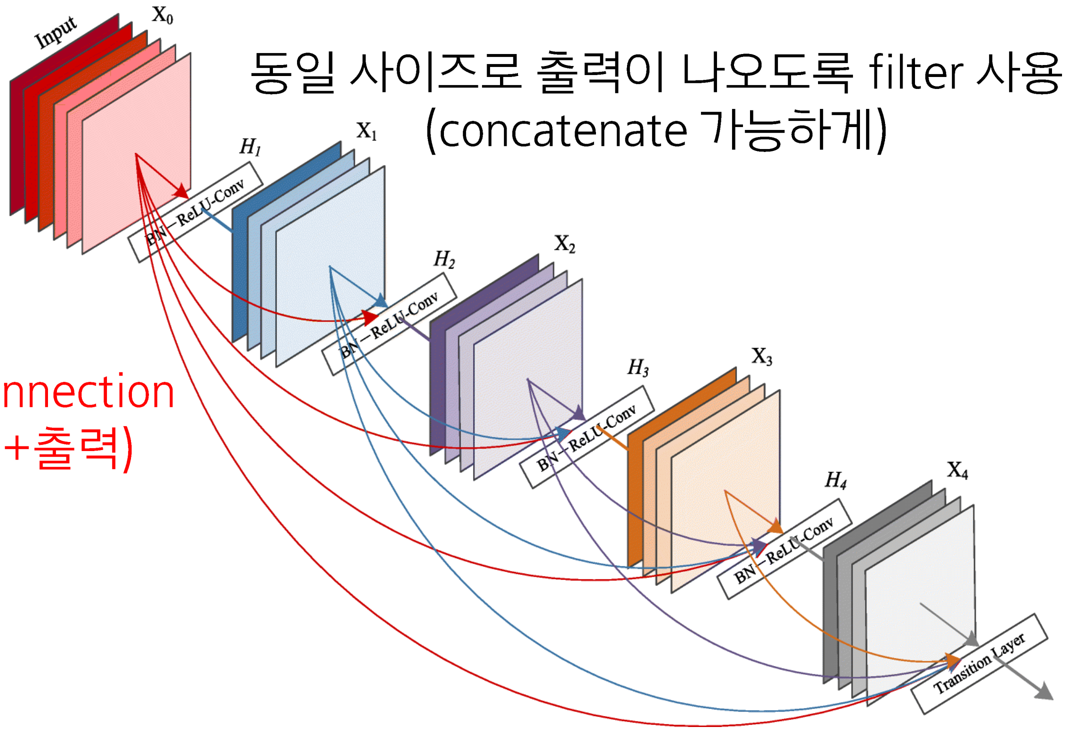
$$(1 \times 1 \times 200) \times (32 \times 32 \times 16) = 3M \qquad (5 \times 5 \times 16) \times (32 \times 32 \times 32) = 1.3M$$

5. Advanced CNNs (ResNet, InceptionNet, DenseNet)

Dense Net (2017)

- Skip connection을 더 dense하게 사용

동일 사이즈로 출력이 나오도록 filter 사용
(concatenate 가능하게)



Skip connection
(입력+출력)

Gradient descent가 더 잘 일어남

5. Advanced CNNs (ResNet, InceptionNet, DenseNet)

성능 비교