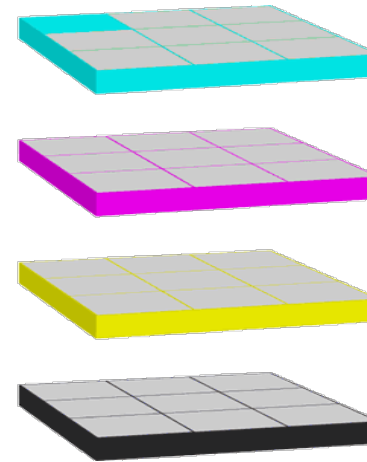
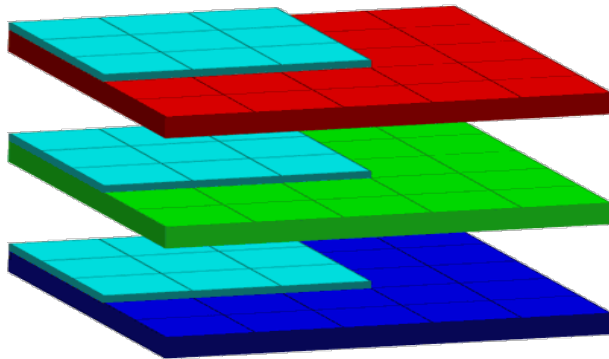


딤러닝 5-6일차

데이터쿵와 고우주
2020.7.24

Convolutional Neural Nets



- Convolution(합성곱)
- 채널(Channel)
- 필터(Filter)
- 커널(Kernel)
- 스트라이드(Strid)
- 패딩(Padding)

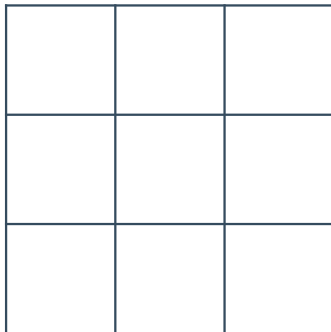
- 피쳐 맵(Feature Map)
- 풀링(Pooling) 레이어

Convolution Setting: Grid Size

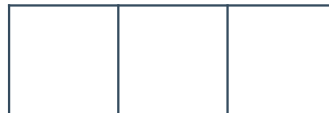
Grid Size (Height and Width):

- 커널(또는 필터)이 한 번에 "인식"하는 픽셀 수
- 일반적으로 홀수를 사용하여 "중심"픽셀이
- 커널은 정사각형과 직사각형

Height: 3, Width: 3



Height: 1, Width: 3



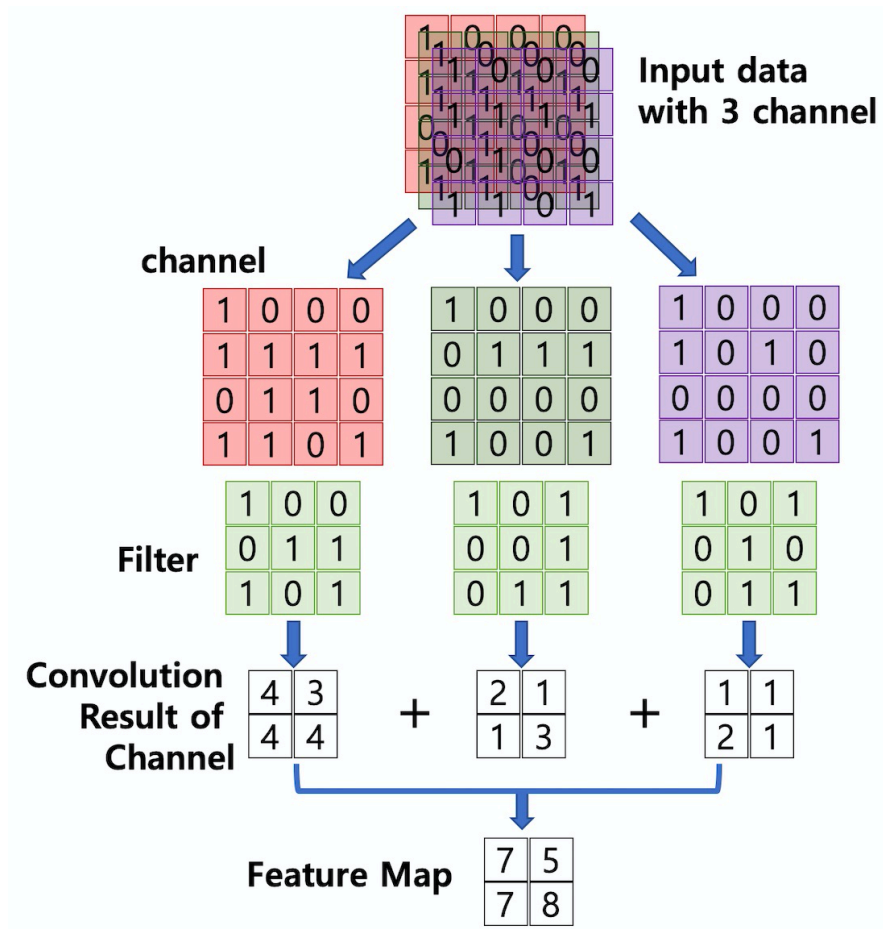
Height: 3, Width: 1



Convolutional Settings—Depth

- 이미지는 종종 각 픽셀 위치와 관련된 여러 개의 숫자가 있습니다.
- 이 숫자를 "채널 " 으로 명칭
 - RGB 이미지: 3 채널
 - CMYK: 4 채널
- 이 채널 수를 "깊이"
- 커널 자체는 입력 채널 수와 동일한 크기의 "깊이"를 가짐
 - EX) RGB 이미지의 5x5 커널
 - $5 \times 5 \times 3 = 75$ 개의 가중치

Convolutional Settings—Depth



- 레이어의 출력에 깊이 설정
- 네트워크는 일반적으로 다양한 커널을 학습
- 각 커널은 각 픽셀 위치에서 단일 숫자 출력
- 레이어에 10개의 커널이 있는 경우 해당 레이어의 출력은 깊이는 10

Convolution Settings: Padding

Padding

- 커널을 직접 사용하면 “가장자리 효과”가 발생
- 주변 픽셀이 충분하지 않기 때문에 가장자리 근처의 픽셀은 "중앙 픽셀"로 미사용
- 패딩은 프레임 주위에 추가 픽셀 추가
- 커널이 이미지를 가로 질러 이동할 때 원본 이미지의 모든 픽셀이 중앙 픽셀
- 추가 된 픽셀의 값은 일반적으로 0으로 채워짐 (패딩 없음)

Without Padding

1	2	0	3	1
1	0	0	2	2
2	1	2	1	1
0	0	1	0	0
1	2	1	1	1

Input

-1	1	2
1	1	0
-1	-2	0

Kernel

-2		

Output

With Padding

0	0	0	0	0	0	0
0	1	2	0	3	1	0
0	1	0	0	2	2	0
0	2	1	2	1	1	0
0	0	0	1	0	0	0
0	1	2	1	1	1	0
0	0	0	0	0	0	0

Input

-1	1	2
1	1	0
-1	-2	0

Kernel

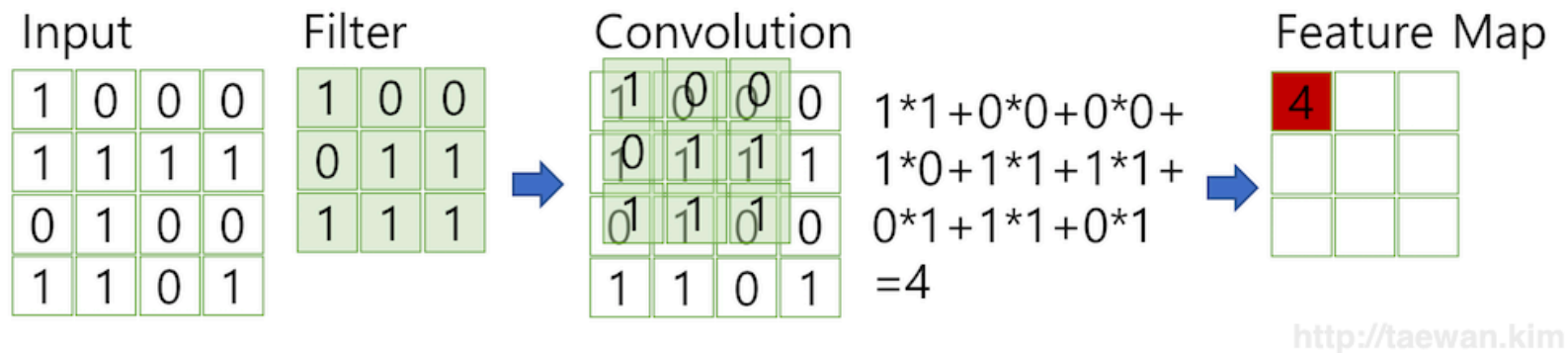
-1				

Output

Convolution Settings: Stride

Stride

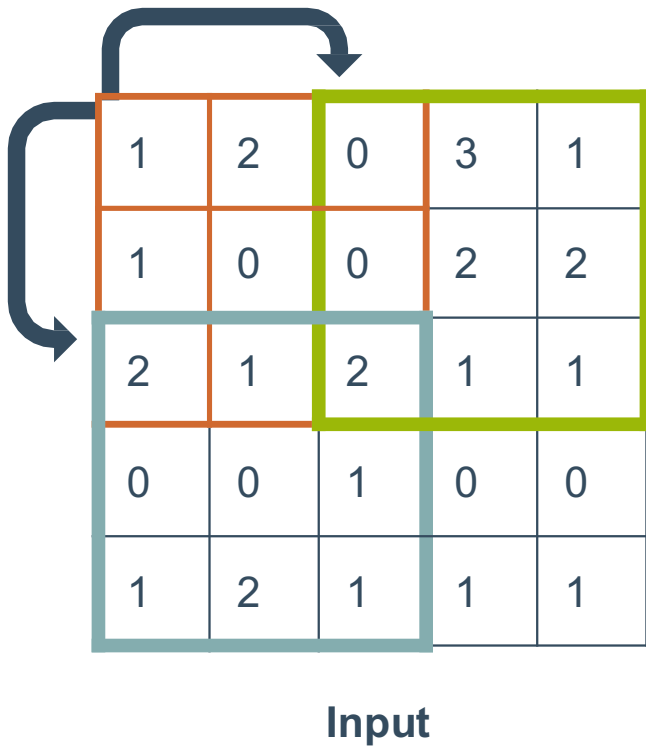
- 커널이 이미지를 가로 질러 이동할 때 "단계 크기"
- 수직 및 수평 단계에 따라 차이 (일반적으로 동일한 값)
- 보폭이 1보다 크면 출력 치수가 축소



Convolution Settings: Stride

- 입력데이터: $W_1 \times H_1 \times D_1$
- Hyper Parameters:
 - 필터크기: F (정방을 가정)
 - 스트라이드: S
- 출력데이터:
 - $W_2 = (W_1 - F)/S + 1$
 - $H_2 = (H_1 - F)/S + 1$
 - $D_2 = D_1$

Stride 2 Example: No Padding



-1	1	2
1	1	0
-1	-2	0

Kernel

-2	3
0	

Output

Stride 2 Example: with Padding

0	0	0	0	0	0	0
0	1	2	0	3	1	0
0	1	0	0	2	2	0
0	2	1	2	1	1	0
0	0	0	1	0	0	0
0	1	2	1	1	1	0
0	0	0	0	0	0	0

Input

-1	1	2
1	1	0
-1	-2	0

Kernel

-1	2	
3		

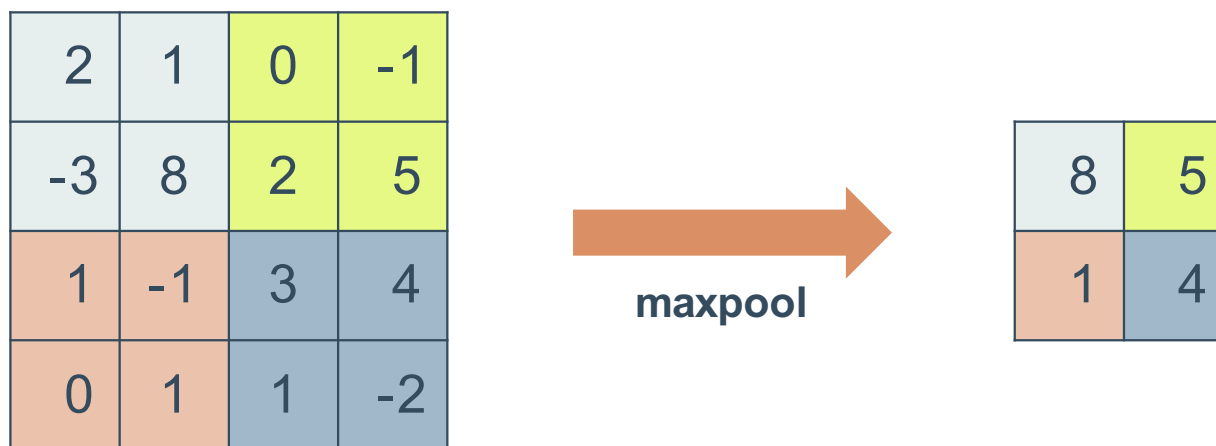
Output

| Pooling

- 픽셀 패치를 단일 값으로 매핑하여 이미지 크기를 줄입니다.
- 학습 대상 파라미터가 없음
- Pooling 레이어를 통과하면 행렬의 크기 감소하지만, 채널 수 변경은 없음
- CNN에서는 주로 Max Pooling을 사용

Pooling: Max-pool

- For each distinct patch, represent it by the maximum
- 2x2 maxpool shown below



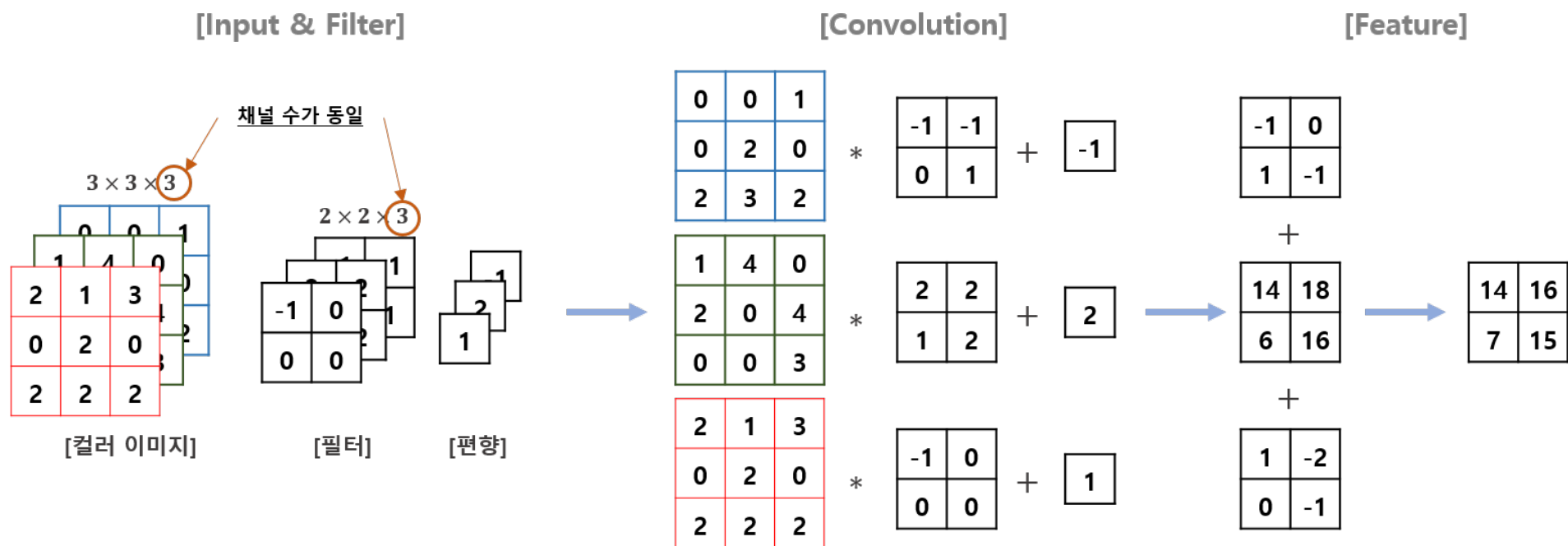
Pooling: Average-pool

- For each distinct patch, represent it by the average
- 2x2 avgpool shown below

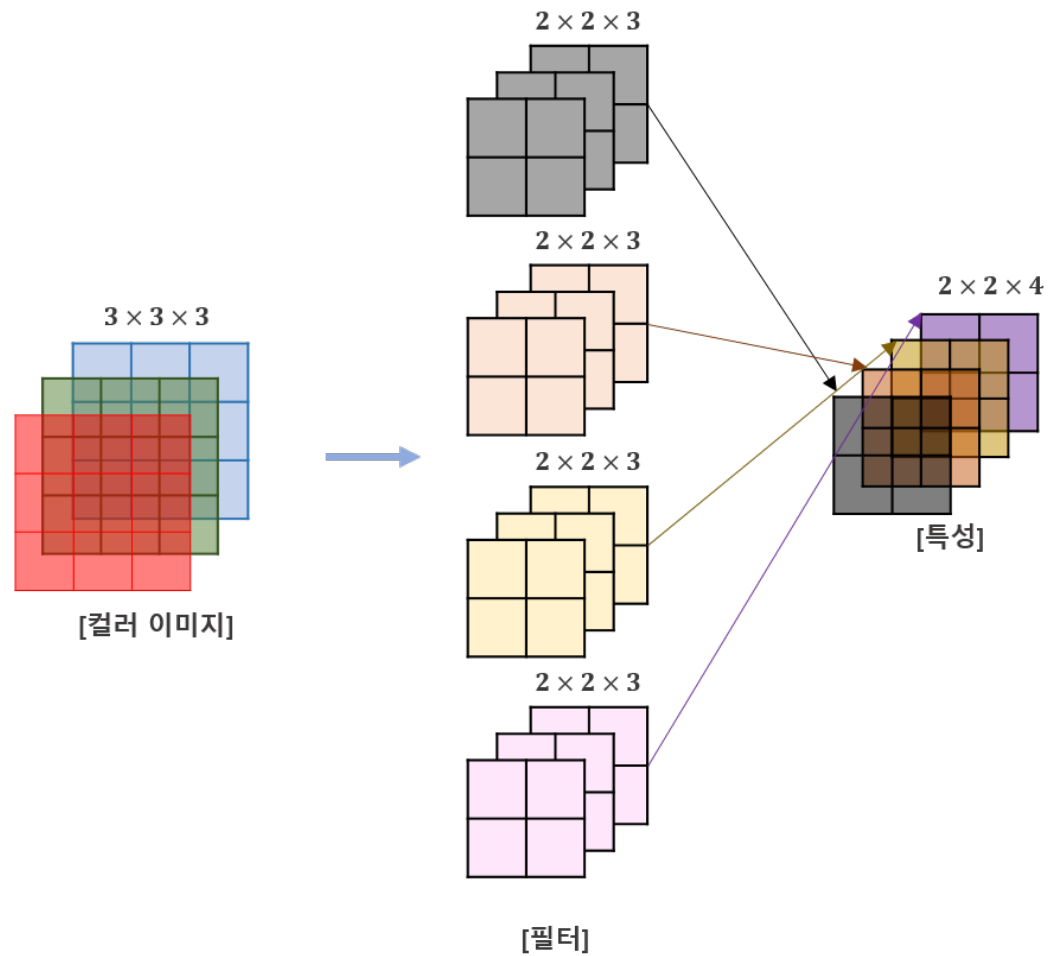
2	1	0	-1
-3	8	2	5
1	-1	3	4
0	1	1	-2



2	1.5
.25	1.5



CNN Architecture



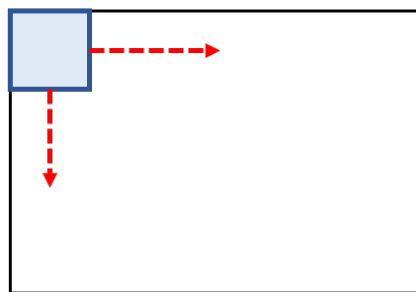
| CNNs 특성

- 입력 데이터: $W_1 \times H_1 \times D_1$ (W_1 : 가로, H_1 : 세로, D_1 : 채널 또는 깊이)
- Hyper Parameters:
 - 필터의 수: K
 - 필터의 크기: F (정방을 가정)
 - 스트라이드: S
 - 패딩: P
- 출력데이터 :
 - $W_2 = (W_1 - F + 2P)/S + 1$
 - $H_2 = (H_1 - F + 2P)/S + 1$
 - $D_2 = K$
- 가중치의 수: $[F^2 \times D_1 + D_1] \times K$

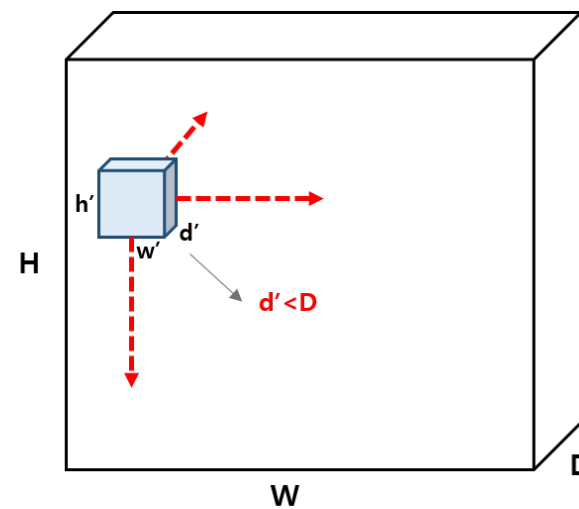
Conv1D, 2D, 3D



[1D Convolution]



[2D Convolution]



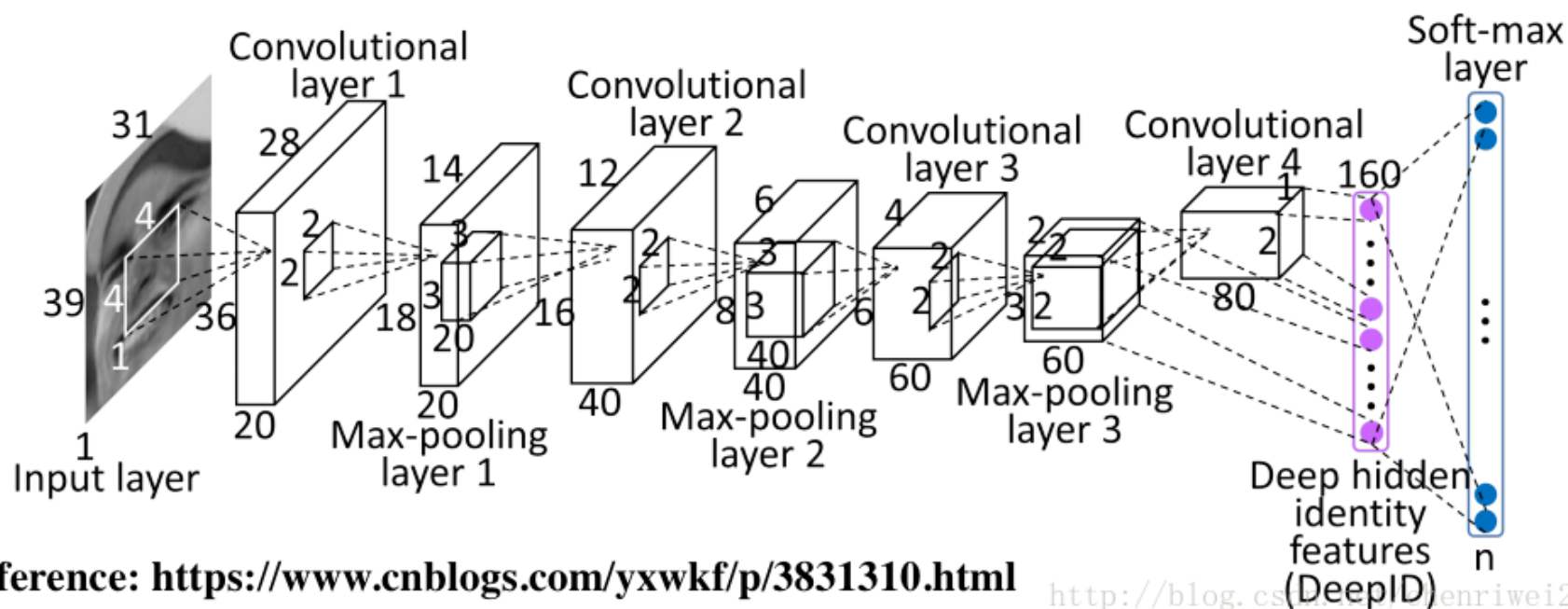
[3D Convolution]

CNN Layers Example

- 입력데이터 Shape: (39, 31, 1)
- 분류 클래스: 100
- 컨볼루션 레이어의 학습 파라미터 수 = “입력 채널수 필터폭 × 필터높이 × 출력 채널수”

Layers	Input channel	Filter	Output channel	Stride	Max pooling	Activation function
Convolution 1	1	(4, 4)	20	1	-	Relu
Max pooling 1	20	-	20	2	(2,2)	-
Convolution 2	20	(3, 3)	40	1	-	Relu
Max pooling 2	40	-	40	2	(2,2)	-
Convolution 3	40	(2, 2)	60	1	1	Relu
Max pooling 3	60	-	60	2	(2,2)	-
Convolution 4	60	(2, 2)	80	1	1	Relu
Flatten	-	-	-	-	-	-
Fully connected	-	-	-	-	-	Softmax

CNN Layers Overview



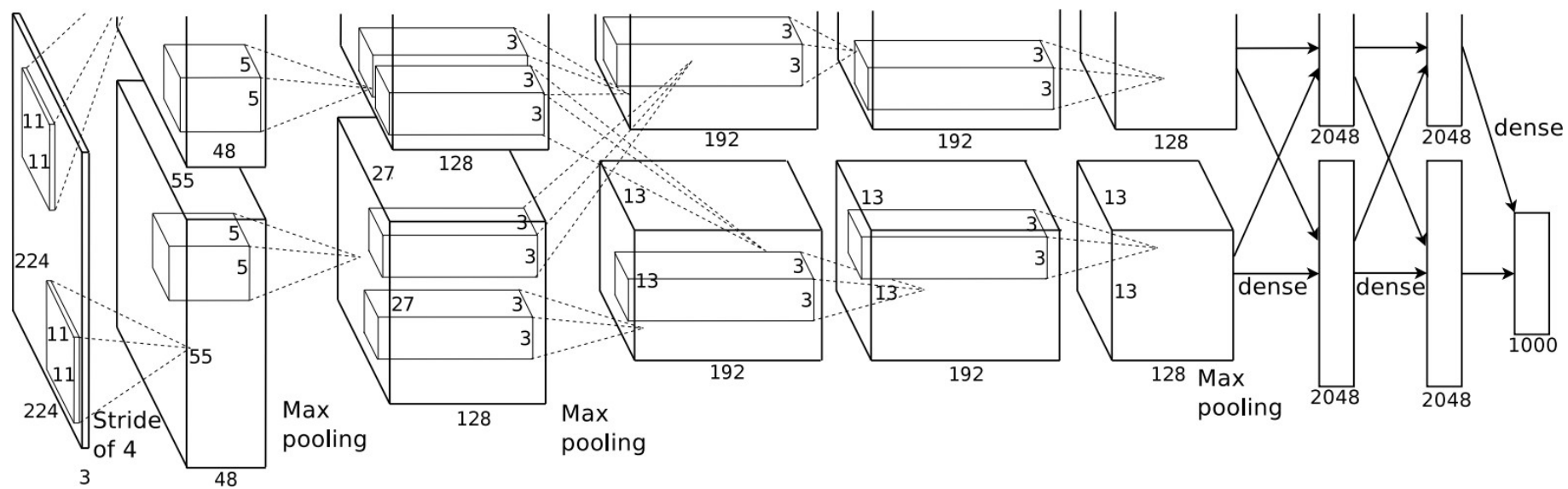
Reference: <https://www.cnblogs.com/yxwxf/p/3831310.html>

<http://blog.csdn.net/chenriwei2>

CNN Layers Overview

$$(n+2p-k) / s$$

Layer	Input channel	Filter	Output channel	Stride	Pooling	활성함수	Input shape	Output shape	파라미터수
Convolution 1	1	(4, 4)	20	1	-	Relu	(39, 31, 1)	(36, 28, 20)	320
Max pooling 1	20	-	20	2	(2, 2)	-	(36, 28, 20)	(18, 14, 20)	0
Convolution 2	20	(3, 3)	40	1	-	Relu	(18, 14, 20)	(16, 12, 40)	7,200
Max pooling 2	40	-	40	2	(2,2)	-	(16, 12, 40)	(8, 6, 40)	0
Convolution 3	40	(2, 2)	60	1	1	Relu	(8, 6, 40)	(6, 4, 60)	21,600
Max pooling 3	60	-	60	(2, 2)	60	-	(6, 4, 60)	(3, 2, 60)	0
Convolution 4	60	(2, 2)	80	1	1	Relu	(3, 2, 60)	(2, 1, 80)	19,200
Flatten	-	-	-	-	-	-	(2, 1, 80)	(160, 1)	0
Fully connected	-	-	-	-	-	Softmax	(160, 1)	(100, 1)	160,000
합계	-	-	-	-	-	Softmax	(160, 1)	(100, 1)	208,320



VGG16 Diagram

