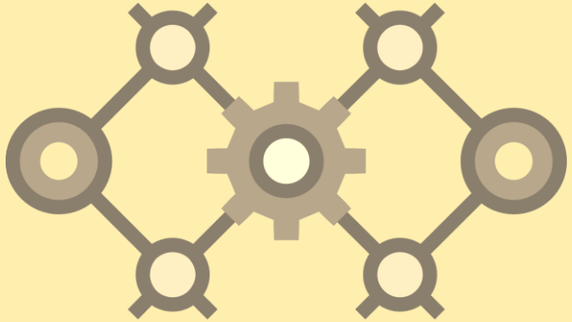
Python을 활용한 이미지 분석

5. 합성곱 신경망

기상청

기상청



합성곱 신경망

1

합성곱 신경망의 개요

1. 합성곱 신경망의 개요

기상청

합성곱 층

합성곱 층의 유래

고양이 실험에서 시작함

두뇌피질이 특정 형태의 그림에서만 반응함

뉴런 사이의 연결 패턴이 **동물의 시각 피질 조직과 유사**하다는 점에서 영감을 얻음

A bit of history:

Hubel & Wiesel,

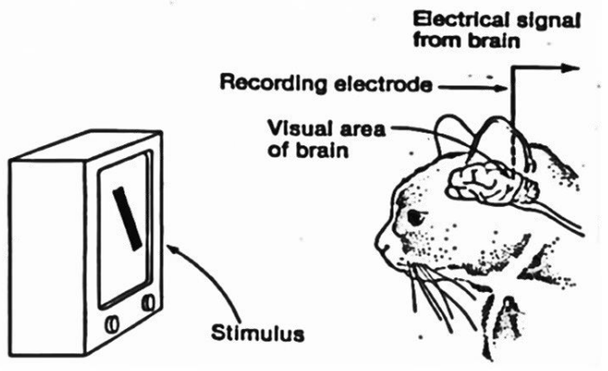
1959

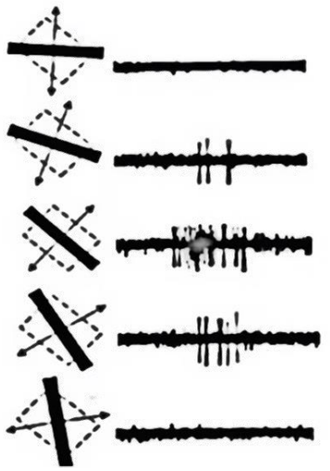
RECEPTIVE FIELDS OF SINGLE NEURONES IN THE CAT'S STRIATE CORTEX

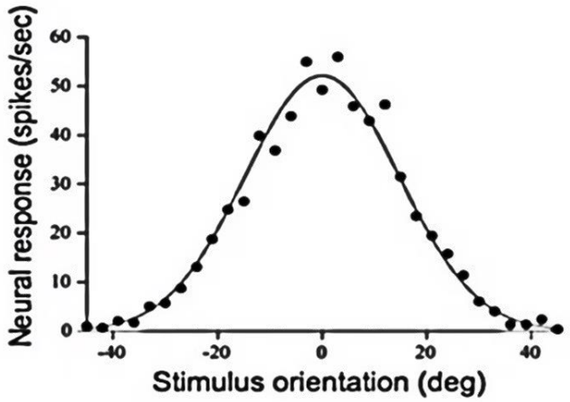
1962

RECEPTIVE FIELDS, BINOCULAR INTERACTION AND FUNCTIONAL ARCHITECTURE IN THE CAT'S VISUAL CORTEX

1968...







[출처 : Stanford University CS231n, http://cs231n.stanford.edu/]

1. 합성곱 신경망의 개요

기상청

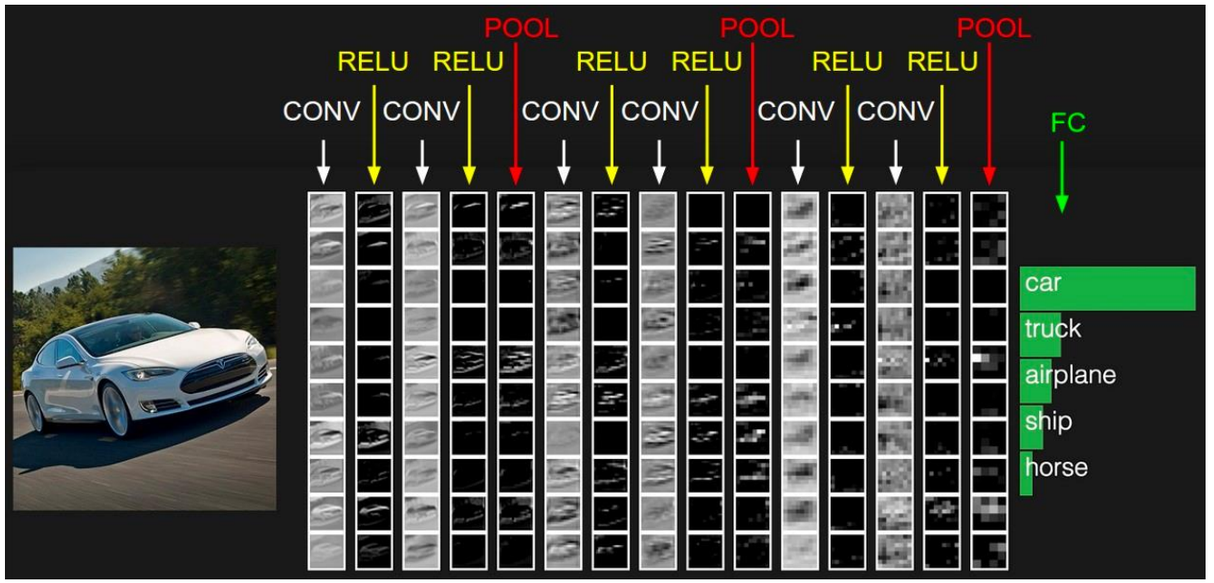
합성곱 신경망

합성곱 신경망의 구성

합성곱 신경망

합성곱 층(Convolution Layer)

풀링 층(Pooling Layer)



[출처 : Stanford University CS231n, http://cs231n.stanford.edu/]

1. 합성곱 신경망의 개요

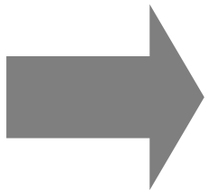
기상청

합성곱 층

완전 연결 계층에서 이미지 문제

2차원 이미지를
1차원으로 변형하며
입력을 넣어줌

0	1	0
0	1	0
0	1	0



0
1
0
0
1
0
0
1
1

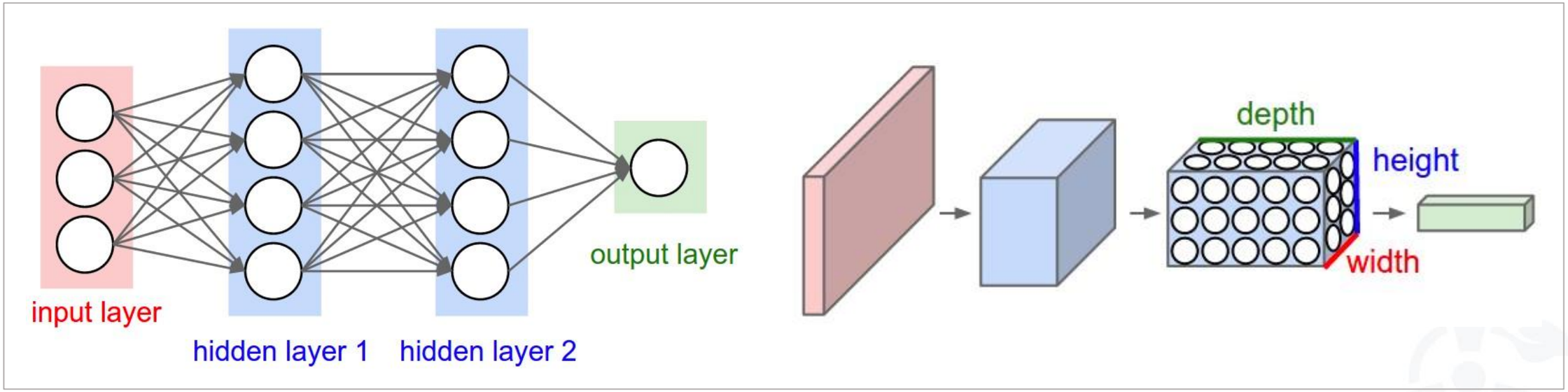
- 공간적 정보를 잃어 **본질적인 패턴 정보가 유실**되는 문제
- 이미지가 커질수록 **파라미터의 수가 증가하며 과적합**되는 문제

1. 합성곱 신경망의 개요

합성곱 층

완전 연결 신경망과 합성곱 신경망

- 완전 연결 신경망에서 이미지를 주입하기 위해서는 1차원 벡터로 평탄화 하는 과정에서 정보의 소실이 발생함
- 합성곱 신경망의 계층에는 너비, 높이, 깊이의 3차원으로 배열된 뉴런으로 이미지를 그대로 주입할 수 있음



[출처 : Stanford University CS231n, http://cs231n.stanford.edu/]

2

합성곱 (Convolution) 층

2. 합성곱(Convolution) 층

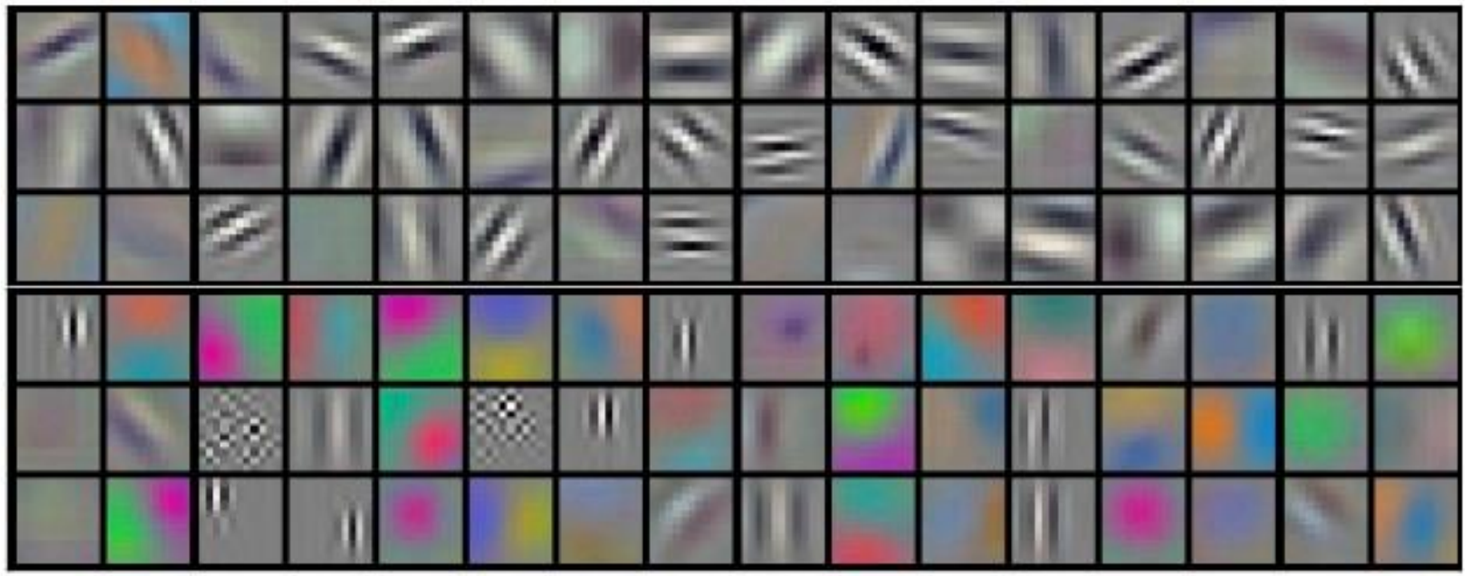
기상청

특징 추출(Feature Extraction)

특징 자동 추출

▶ 이미지에서 부분마다의 특징을 자동으로 추출함

합성곱	• 특정한 패턴의 특징이 어디에서 나타나는지를 확인하는 도구 • 영어로 컨볼루션(Convolution)으로 불림



[출처 : Stanford University CS231n, https://cs231n.github.io/]

2. 합성곱(Convolution) 층

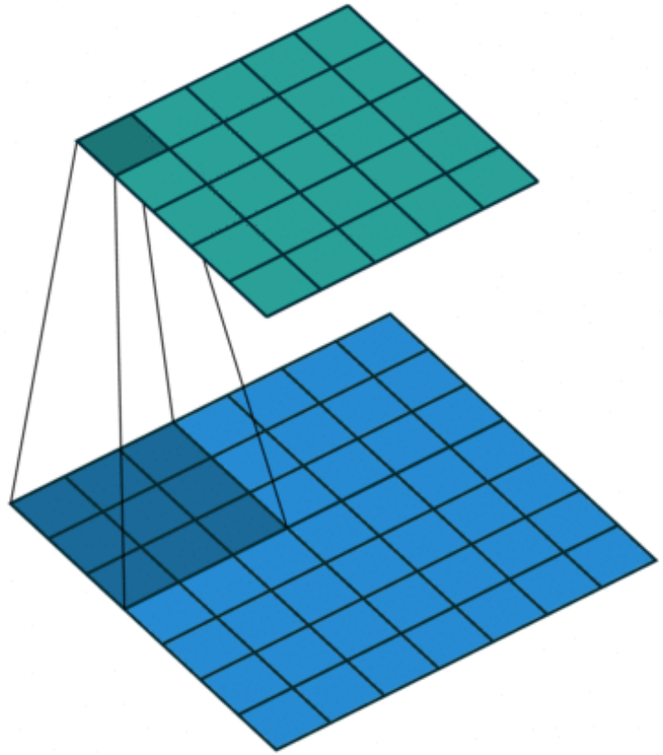
기상청

합성곱 연산

커널(Kernel) 또는 필터(Filter)라는 n x m 크기의 행렬로 높이와 너비의 곱하기 크기의 이미지를 처음부터 끝까지 겹치며 이동함

겹쳐지는 부분의 각 이미지와 커널의 원소의 값을 곱해서 모두 더한 값을 출력함

- 이미지 가장 왼쪽부터 오른쪽까지 순차적으로 이동함
- 서로 다른 뉴런이 전체 시야를 볼 수 있도록 부분적으로 중첩되는 원리로 구현함

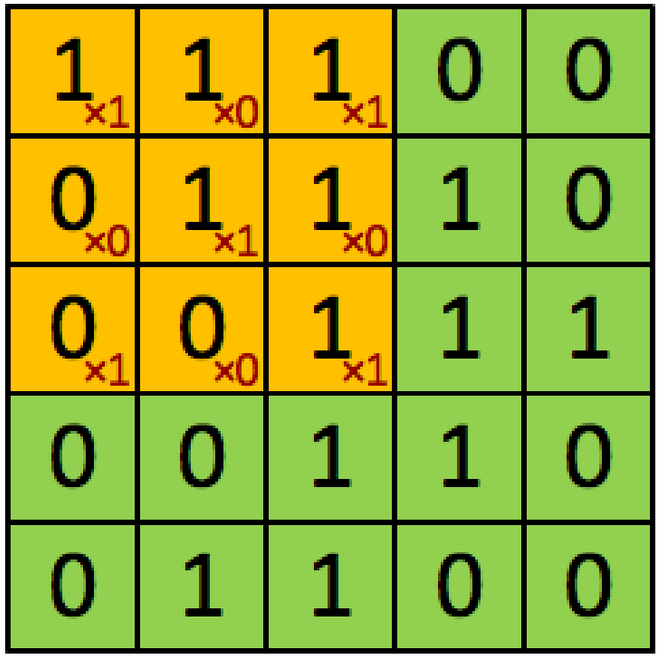


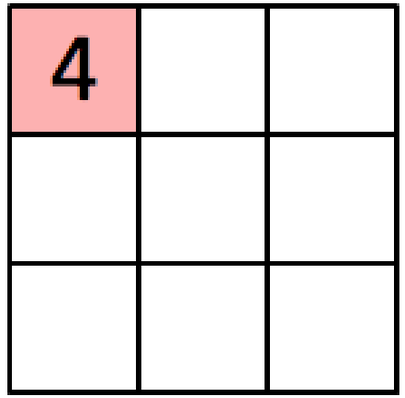
[출처 : 위키미디어, https://commons.wikimedia.org/]

2. 합성곱(Convolution) 층

기상청

합성곱 연산

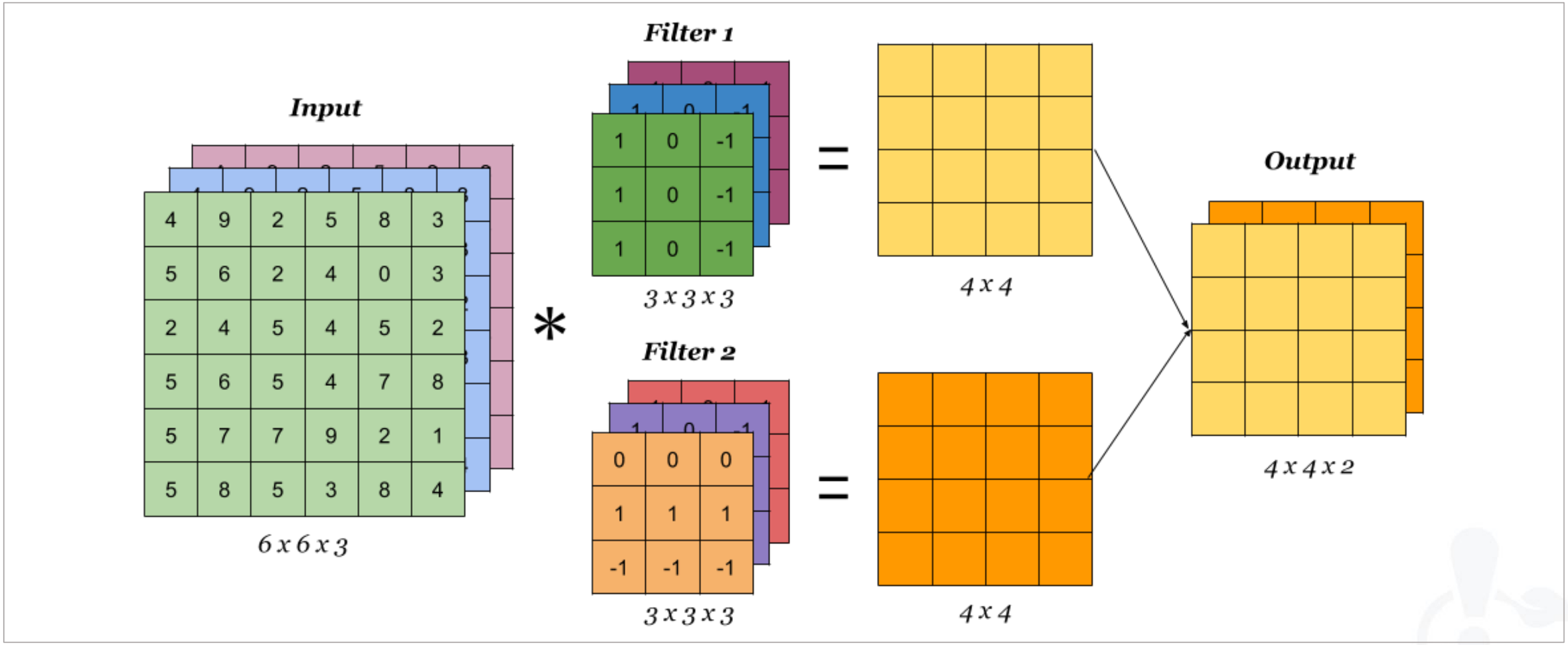




[출처 : ttp://ufldl.stanford.edu/tutorial/supervised/FeatureExtractionUsingConvolution/]

2. 합성곱(Convolution) 층

합성곱 과정



[출처 : https://indoml.com/2018/03/07/student-notes-convolutional-neural-networks-cnn-introduction/]

2. 합성곱(Convolution) 층

기상청

합성곱 연산

합성곱 과정

- 데이터의 특징을 추출하는 과정

데이터에 각 성분의 인접 성분들을 조사해 특징을 파악함

→

파악한 특징을 한장으로 도출시키는 과정

→

이미지의 특정부분을 추상화 하여 특정 층으로 표현함

→

하나의 압축과정으로 파라미터의 갯수를 효과적으로 줄여줌

[Image]

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

✚

[Filter]

1	0	1
0	1	0
1	0	1

➡

[Convolved Feature]

4	3	4
2	4	3
2	3	4

2. 합성곱(Convolution) 층

기상청

기본 용어 정의

커널 (Kernel)	• 슬라이딩 윈도우 형식으로 입력된 데이터 상에서 좌에서 우로, 위에서 아래로 내적 연산을 진행함
필터 (Filter)	• 작은 필터를 여러 개 중첩하면 원하는 특징을 더 돋보이게 연산량을 줄일 수 있음 • 입력 데이터의 채널 수와 필터의 채널 수는 같아야 함 • RGB 색이 추가된다면 총 3개의 채널을 갖게 되며 커널은 여전히 n x n x 1 개의 텐서이지만, 필터의 차원은 m x m x 3이 되어 3개의 커널을 포함하게 됨

2. 합성곱(Convolution) 층

기상청

기본 용어 정의

스트라이드 (Stride)	• 커널의 이동 보폭 1이라면 한 픽셀씩 이동한다는 의미 • 큰 값을 입력하면 계산 단위가 줄어듦
피처맵 (Feature Map)	• 입력과 커널의 합성곱 연산 결과

2. 합성곱(Convolution) 층

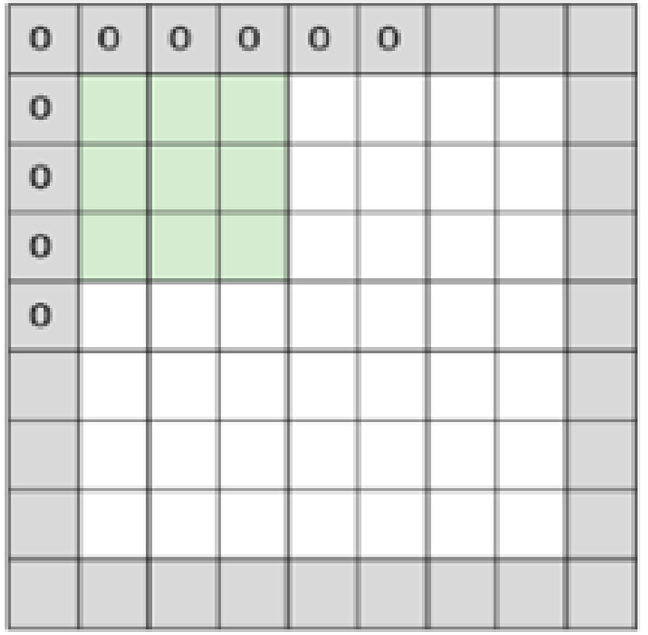
기상청

패딩(Padding)

패딩의 목적

합성곱 연산을 수행하기 전에 입력 데이터 주변을 0,1 등의 특정 값을 사용해서 채우는 것

- 0으로 채우는 Zero-Padding을 주로 사용함
- 입력 데이터와 출력 데이터의 크기를 맞추기 위해서 사용함
- 입력 데이터에 필터를 적용하여 합성곱 연산 수행시 이미지 데이터가 축소되는 것을 방지함
- 모서리 부분의 데이터를 충분히 활용하기 위해 사용함



[출처 : Stanford University CS231n, http://cs231n.stanford.edu/]

2. 합성곱(Convolution) 층

기상청

패딩(Padding)

패딩의 종류

Valid Padding

- Padding 하지 않는 것을 의미함
- 입력보다 출력의 크기가 작아짐

Same Padding

- 입력과 출력 이미지의 크기가 같음
- Zero Padding을 의미함

▶ tf.keras에서는 Conv2D에 Padding 옵션을 제공하고 있음

[Image(+Zero Padding)]

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

2. 합성곱(Convolution) 층

기상청

합성곱 함수 이해

tf.keras.layers.Conv2D

filters	필터의 갯수	padding	Zero-Padding의 유무
kernel_Size	필터의 크기	activation	활성화 함수
strides	이동할 픽셀 크기		

```
tf.layers.Conv2D(filters=64,
                 kernel_size=(3, 3),
                 activation='relu')
```

2. 합성곱(Convolution) 층

기상청

실습

예제

문제 상황	실습 코드
합성곱 신경망 구성	tf.keras.layers.Conv2D()

예시화면

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D

model = Sequential()
model.add(Conv2D(filters=32, kernel_size=(3, 3), strides=1, padding='same', input_shape=(32,32, 3)))
print(model.output_shape)
model.add(Conv2D(filters=32, kernel_size=(3, 3), strides=1, padding='valid',))
print(model.output_shape)
```

3

풀링(Pooling) 층

3. 풀링(Pooling) 층

기상청

풀링의 과정

풀링(Pooling)

▶ 개념

풀링 (Pooling)	합성곱 과정을 거친 층의 사이즈를 줄여주는 과정

▶ 장점

01	단순히 데이터의 사이즈를 줄여주어 학습 속도를 높여줌
02	노이즈를 상쇄시켜 줌
03	미세한 부분에서 일관적인 특징을 제공하기 때문에 오버피팅을 억제해 줌

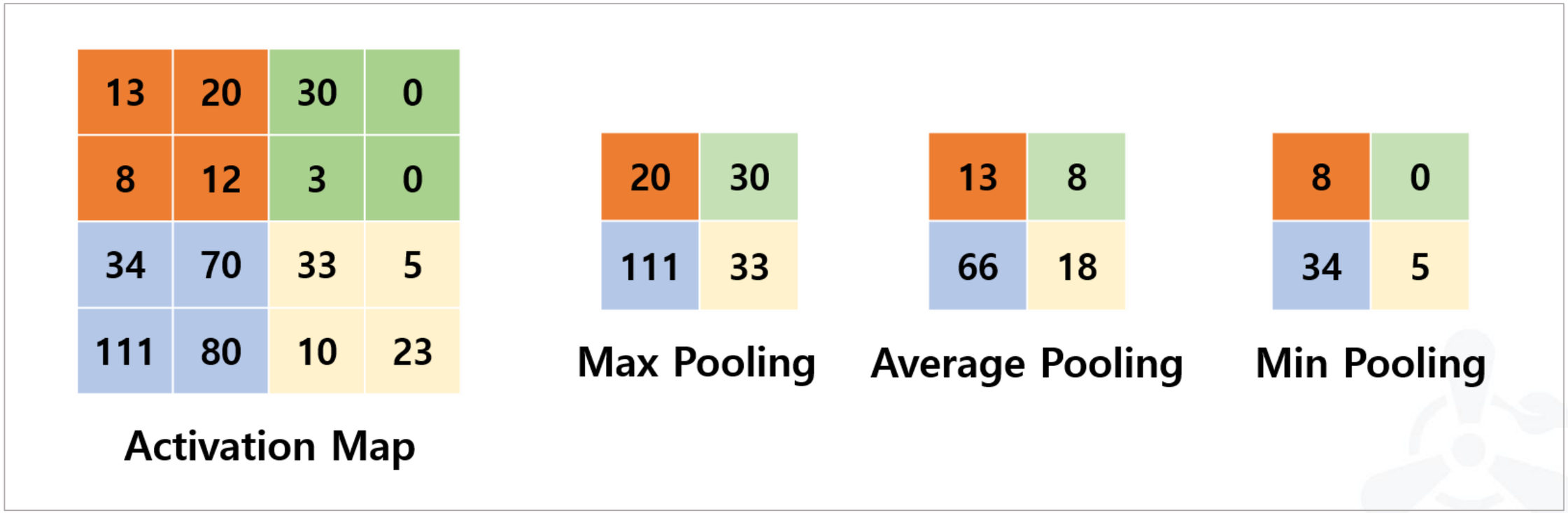
3. 풀링(Pooling) 층

기상청

풀링의 종류

Max, Average, Min Pooling

- 대상 영역에서 최댓값, 평균, 최솟값 등을 취하는 것에 따라 풀링의 종류가 달라짐
- 보통은 Max Pooling을 가장 많이 사용함

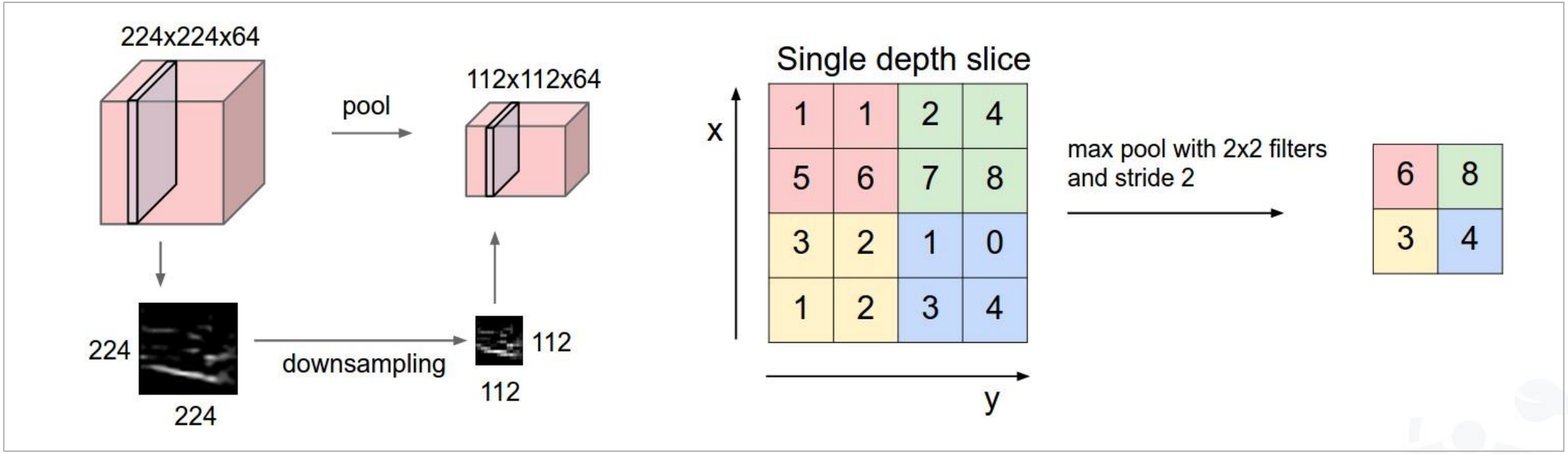


3. 풀링(Pooling) 층

풀링의 장점

01 전체 데이터의 사이즈가 줄어들어 연산에 들어가는 컴퓨팅 리소스가 줄어듦

02 데이터의 크기를 줄이며 소실이 발생하기 때문에 오버피팅을 방지해 줌



[출처 : Stanford University CS231n, http://cs231n.stanford.edu/]

3. 풀링(Pooling) 층

기상청

풀링 함수의 이해

tf.keras.layers.MaxPool2D

pool_Size

Pooling에 사용할 Filter의 크기를 정하는 것(단순한 정수 또는 튜플형태(N,N))

strides

Pooling에 사용할 Filter의 Strides를 정하는 것

padding

"valid"(Padding을 하지 않음) or "same"(결과 Size가 Input Size와 동일하게 Padding)

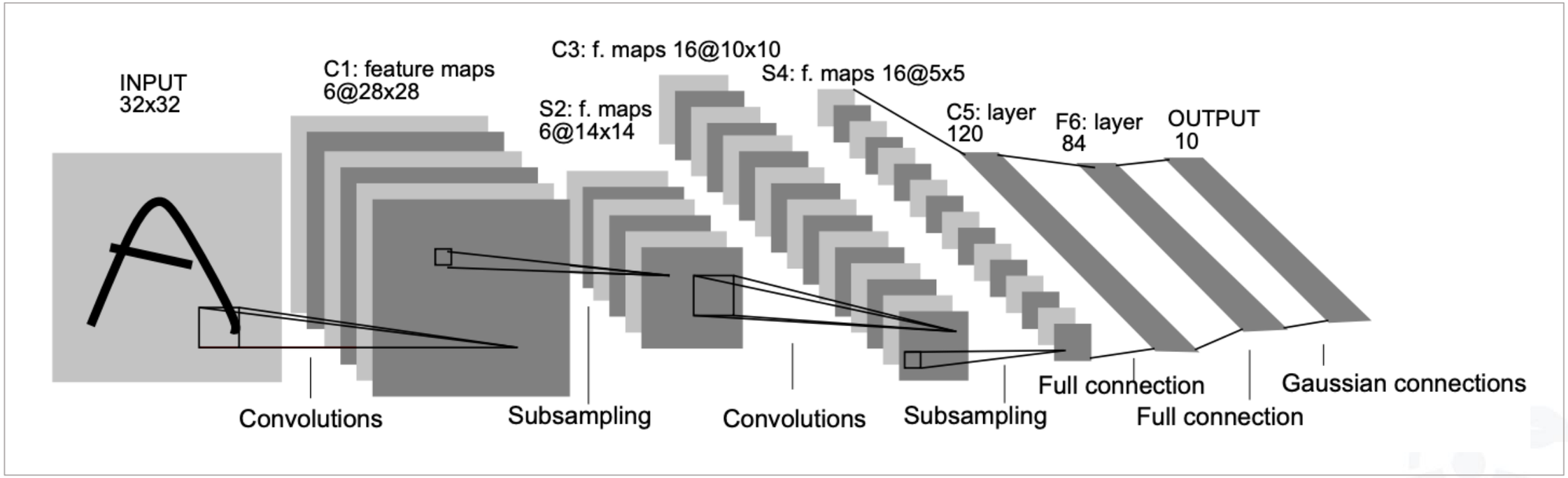
```
tf.keras.layers.MaxPool2D(
    pool_size=(2, 2),
    strides=(1, 1),
    padding='same',
    data_format=None,
    **kwargs
)
```

3. 풀링(Pooling) 층

합성곱 신경망 구성

LeNet-5

- 얀 르쿤(Yann LeCun) 교수에 의해 만들어짐
- 합성곱과 풀링 과정 후에 완전연결을 사용하여 결과를 만들어 내는 방식



[출처 : http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf]

3. 풀링(Pooling) 층

기상청

실습

예제

문제 상황	실습 코드
합성곱 신경망 구성	tf.keras.layers.MaxPooling2D()

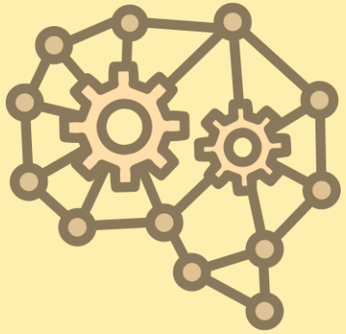
예시화면

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Dense

model = Sequential()
model.add(Conv2D(filters=32, kernel_size=(3, 3), strides=1, padding='same', input_shape=(150,150, 3),))
model.add(MaxPooling2D(2,2))
model.add(Conv2D(filters=32, kernel_size=(4, 4), strides=1, padding='valid',))
model.add(MaxPooling2D(2,2))
print(model.output_shape)
model.add(Flatten())
print(model.output_shape)
model.add(Dense(256, activation='relu'))
model.add(Dense(5, activation='softmax'))
```

강의 마무리

기상청



KEY POINT

- 합성곱 신경망의 개요
  - 3차원으로 배열된 뉴런으로 이미지를 그대로 주입할 수 있음
  - 합성곱 연산과 서브샘플링을 적용하는 것을 반복함
  - 마지막에 완전 연결층을 통해 데이터를 분류함
- 합성곱(Convolution) 층
  - 특정한 패턴의 특징이 어디에서 나타나는지를 확인하는 도구
  - 필터의 갯수, 크기, Strides로 이동할 픽셀의 크기, 패딩 유무, 활성화 함수 등으로 구성됨
- 풀링(Pooling) 층
  - 합성곱 과정을 거친 층의 사이즈를 줄여주는 과정
  - 데이터의 사이즈를 줄여주기 때문에 빠르게 학습하는데 도움이 됨
  - 노이즈를 상쇄시키고 오버피팅을 억제시켜 줌

python™

Python을 활용한 이미지 분석

5. 합성곱 신경망

"이번 시간을 모두 마치셨습니다.
수고하셨습니다."

기상청