

14장 합성곱신경망: 컴퓨터비전

감사의 글

자료를 공개한 저자 오렐리앙 제롱과 강의준비에 필요한 자료를 지원한 한빛아카데미에게 진심어린 감사를 전합니다.

- 합성곱신경망(Convolutional Neural Network, CNN)
 - 대뇌의 시각피질 연구에서 출발하여 1980년대부터 이미지인식 분야에서 사용됨.
 - 일부 복잡한 이미지 처리문제에서 사람의 능력 초과함.
 - 주요 활용 예제: 이미지검색 서비스, 자율주행 자동차, 영상 자동분류 등
 - 음성인식, 자연어 처리 등 다양한 분야에서도 활용됨.

주요 내용

- CNN의 구성요소
- TF와 케라스를 이용한 CNN의 구현
- 가장 뛰어난 성능의 CNN 구조 살펴보기
- 활용예제
 - 물체탐지(object detection)
 - 이미지 상에서 여러 물체 구분하고 물체 주위에 경계상자(bounding box) 그리기
 - 의미분할(semantic segmentation, 시맨틱 분할)
 - 서로 다른 물체에 속한 픽셀 분류하기

시각피질 구조와 합성곱 신경망

- 시각피질 안의 많은 뉴런이 작은 국부수용장(local receptive field)를 가진 것으로 보임.

A bit of history:

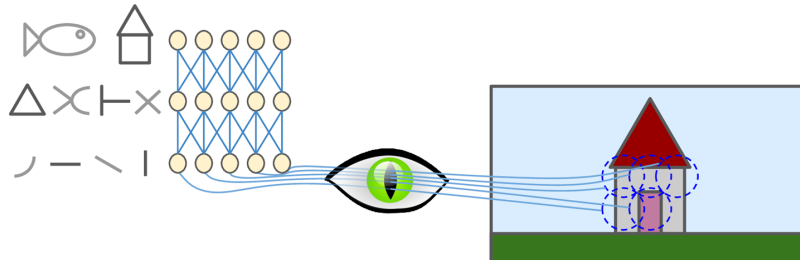
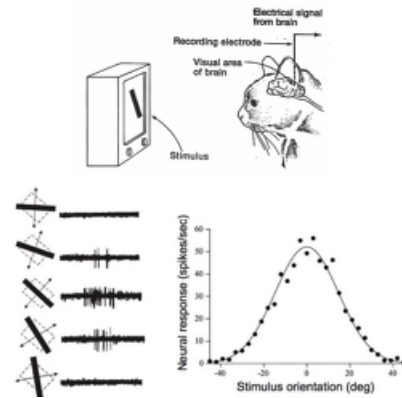
Hubel & Wiesel,
1959

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

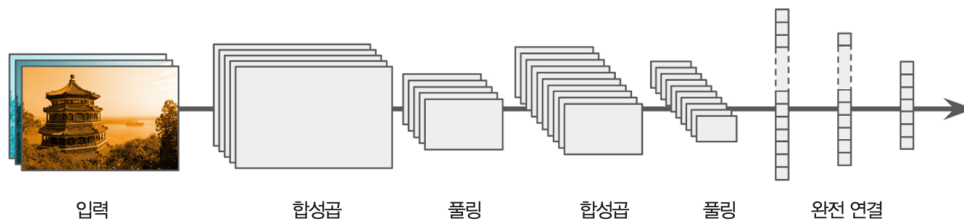
1962

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

1968...



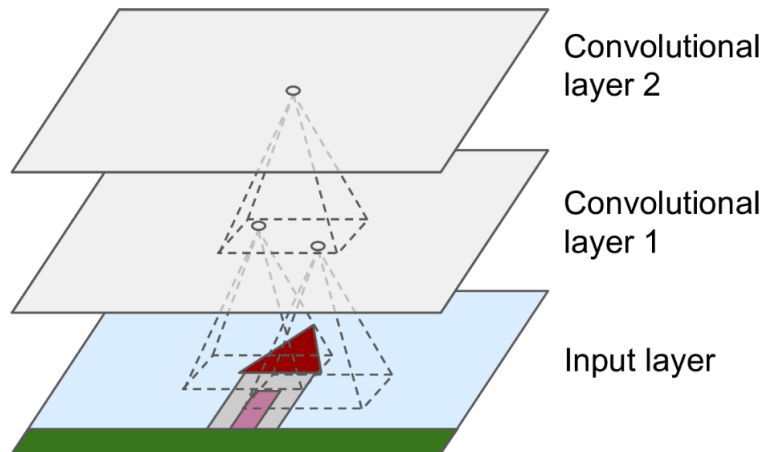
- 합성곱신경망(CNN)으로 발전
 - 합성곱 층(convolution layer)
 - 풀링 층(pooling layer)



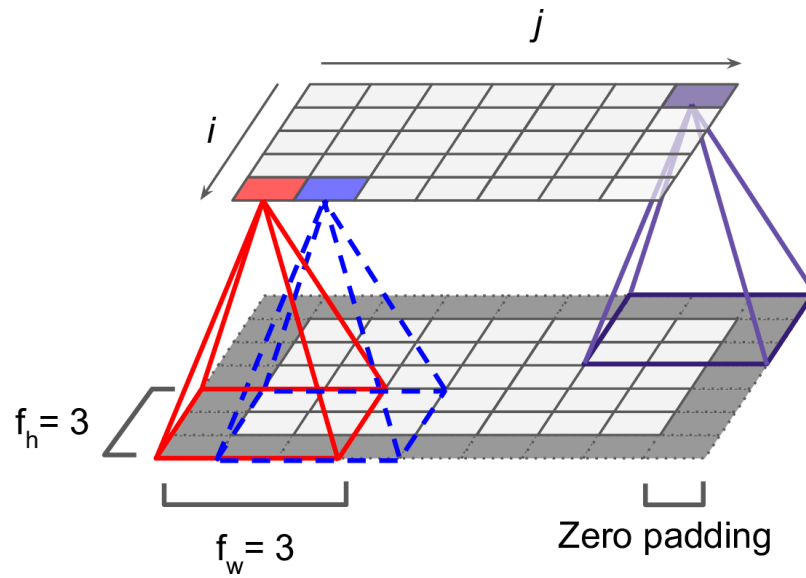
합성곱 층(Convolutional layer)

- CNN의 가장 중요한 구성요소

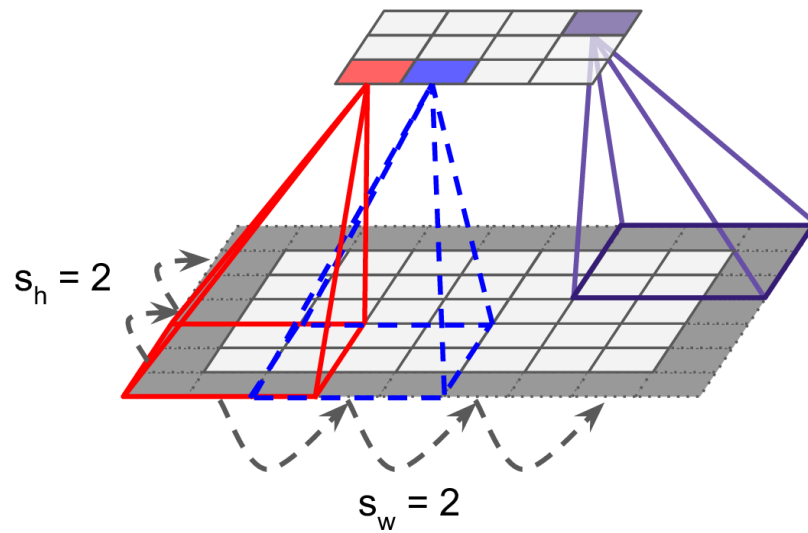
- 사각 형태의 국부수용장을 모방한 CNN 층



- 층과 제로패딩 사이의 연결



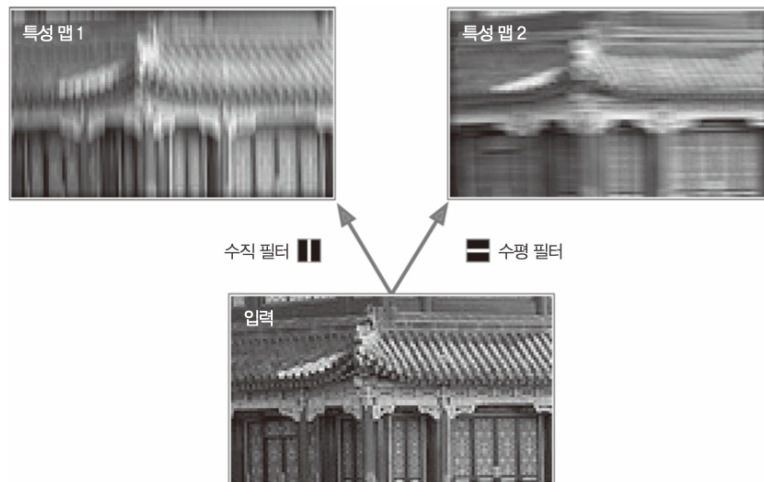
- 보폭(스트라이드, stride) 2를 사용한 차원축소 지원 CNN 층



필터 (합성곱 커널)

- 뉴런에 사용되는 가중치
- 사용되는 수용장과 동일한 모양의 이미지로 표현됨.

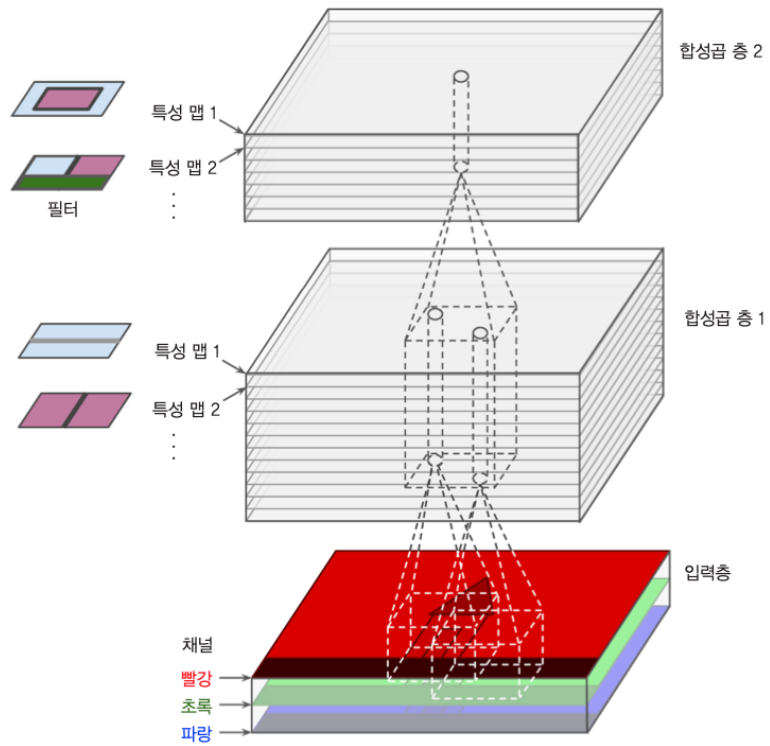
- 다양한 필터 사용
 - 아래 그림은 필터 두 개의 활용을 보여줌.



특성지도(feature map)

- 특성지도: 필터 각각을 사용하여 생성된 출력값
- 필터마다 사용되는 가중치와 편향이 다름.

- 수십, 수백 개의 필터를 사용함.

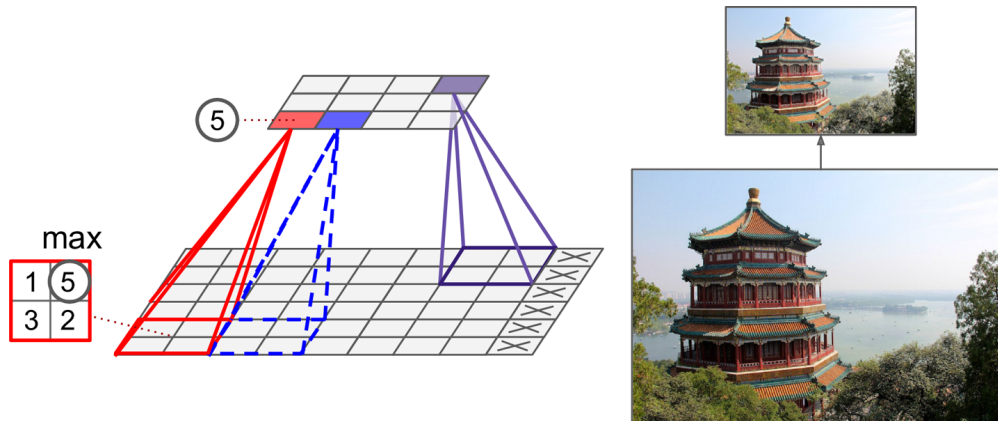


풀링 층(pooling layer)

- 계산량과 메모리 사용량을 줄이면서 과대적합의 위험도를 줄여주는 용도로 사용됨.
- 가중치 사용하지 않지만, 보폭(stride)를 사용하여 차원을 축소시키는 기능 수행.

최대 풀링 층

- 예제: 2×2 풀링 커널, 보폭 2, 패딩 없음



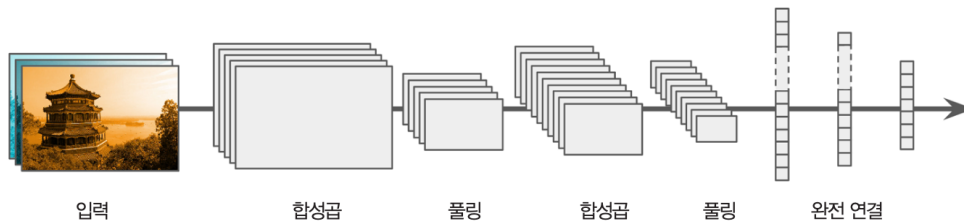
평균 풀링 층

- 풀링커널 구역내의 평균값 활용

전역 평균 풀링 층

- 각 특성지도의 평균 활용

CNN 구조



케라스 활용: 패션 MNIST

```
model = keras.models.Sequential([
    keras.layers.Conv2D(64, 7, activation="relu", padding="same",
        input_shape=[28, 28, 1]), keras.layers.MaxPooling2D(2),
    keras.layers.Conv2D(128, 3, activation="relu", padding="same"),
    keras.layers.Conv2D(128, 3, activation="relu", padding="same"),
    keras.layers.MaxPooling2D(2),
    keras.layers.Conv2D(256, 3, activation="relu", padding="same"),
    keras.layers.Conv2D(256, 3, activation="relu", padding="same"),
    keras.layers.MaxPooling2D(2),
    keras.layers.Flatten(), keras.layers.Dense(128, activation="relu"),
    keras.layers.Dropout(0.5), keras.layers.Dense(64, activation="relu"),
    keras.layers.Dropout(0.5), keras.layers.Dense(10, activation="softmax")
])
```