

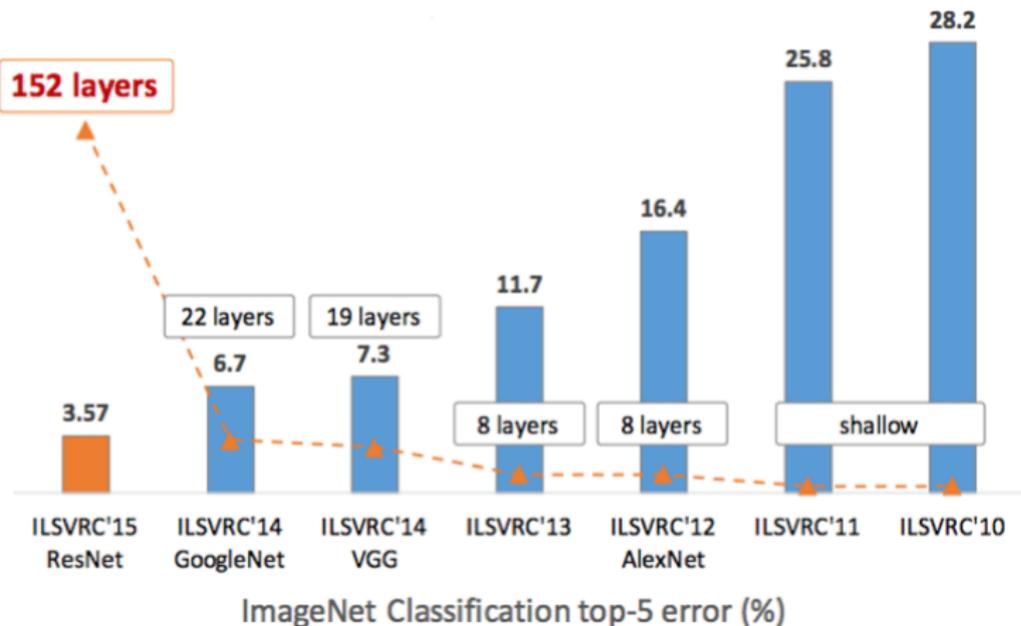
In-Class 20: Convolutional Neural Network

[SCS4049] Machine Learning and Data Science

Seongsik Park (s.park@dgu.edu)

AI Department, Dongguk University

ImageNet challenge



The architecture of the visual cortex

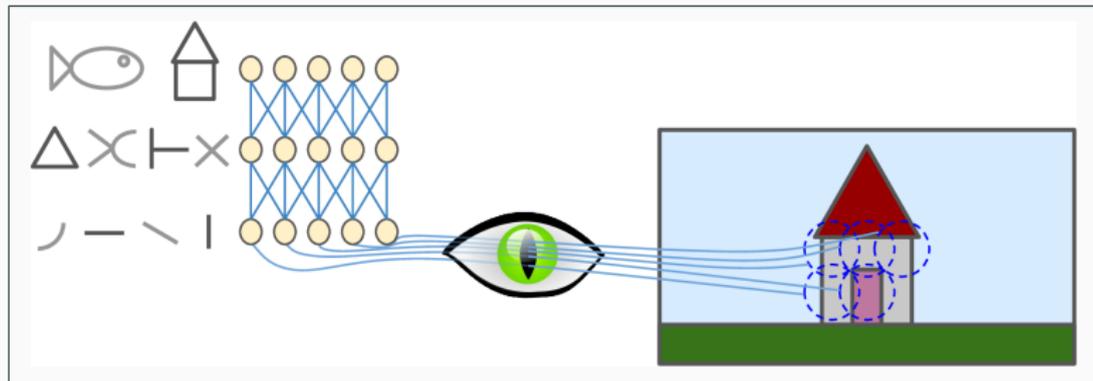


Figure 14-1. Local receptive fields in the visual cortex

In particular, they showed that many neurons in the visual cortex have a small **local receptive field**, meaning they react only to visual stimuli located in a limited region of the visual field (see Figure 14-1, in which the local receptive fields of five neurons are represented by dashed circles). The receptive fields of different neurons may overlap, and together they tile the whole visual field.

Convolutional layer

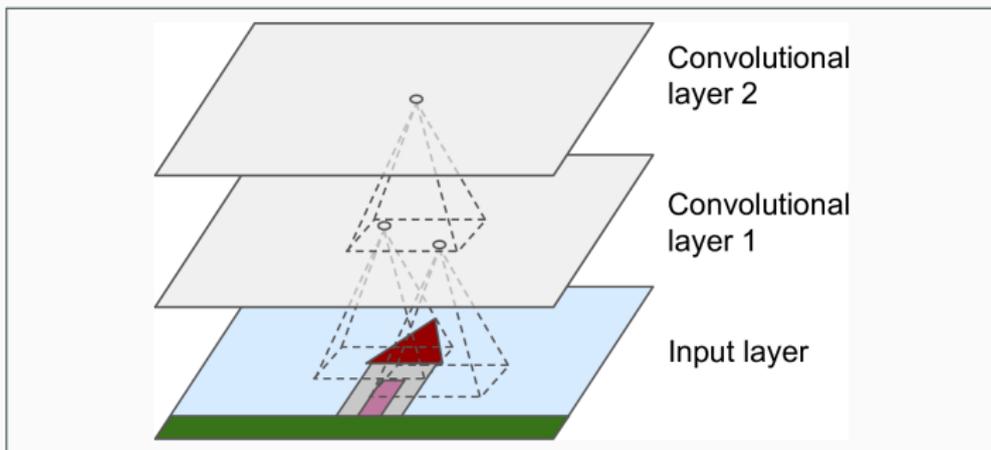


Figure 14-2. CNN layers with rectangular local receptive fields

The most important building block of a CNN is the **convolutional layer**: 6 neurons in the first convolutional layer are not connected to every single pixel in the input image (like they were in previous chapters), but only to pixels in their receptive fields (see Figure 14-2). In turn, each neuron in the second convolutional layer is connected only to neurons located within a small rectangle in the first layer. This architecture allows the network to concentrate on small low-level features in the first hidden layer, then assemble them into larger higher-level features in the next hidden layer, and so on.

Convolutional neural network

- 컨볼루션 신경망(CNN)은 다층 신경망(multi-layer neural networks)의 특별한 형태
- CNN은 시각 피질의 국소 수용역(Local Receptive Field, LRF)과 시신경 세포의 방향 선택적 성격의 발견에 의해서 신경 생물학적인 영감으로 만들어짐
- 내재적이면서 자동적으로 유의미한 특징(feature)들을 추출하는 신경망 구조
- CNN은 이미지로부터 위상학적 성질들을 추출할 수 있는 피드포워드(feed-forward) 신경망
- 다른 모든 신경망들처럼 CNN은 변형된 역전파 알고리즘을 사용하여 학습
- CNN은 최초의 전처리 만으로도 픽셀 이미지들로부터 시각적 패턴을 직접 인식할 수 있도록 설계
- 극도로 변화가 많은 패턴들도 인식 가능함 (예: 손글씨)

Convolution

Given a filter kernel \mathcal{H} , the convolution of the kernel with image \mathcal{F} is an image \mathcal{R} . The (i, j) -th component of \mathcal{R} is given by

$$R_{ij} = \sum_{u,v} H_{i-u, j-v} F_{uv}. \quad (1)$$

- **kernel** of the filter: the pattern of weights used for a linear filter
- **convolution**: the process of applying the filter

This operation is called **convolution**

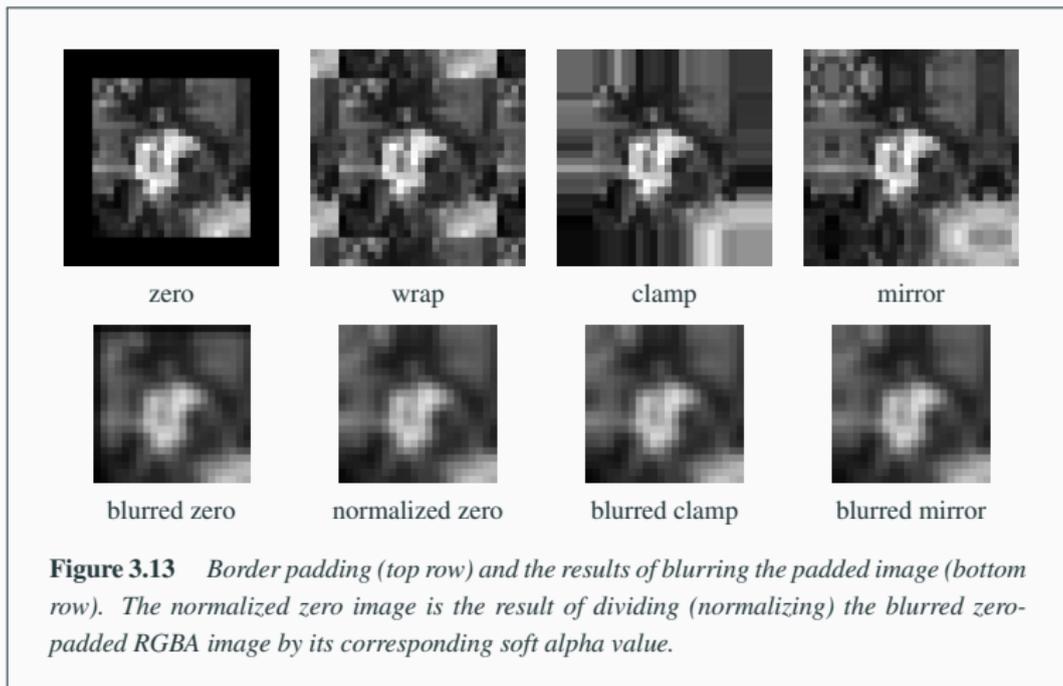
$$R(f) = (h * f) \quad (2)$$

- *commutative*: $(g * h)(x) = (h * g)(x)$
- *associative*: $(f * (g * h)) = ((f * g) * h)$
- *distributive*: $f * (g + h) = f * g + f * h$

Padding (border effects)

- **zero**: set all pixels outside the source image to 0
- **constant**: set all pixels outside the source image to a specified border value
- **clamp**: repeat edge pixels indefinitely
- **wrap**: loop “around” the image in a “toroidal” configuration
- **mirror**: reflect pixels across the image edge
- **extend**: extend the signal by subtracting the mirrored version of the signal from the edge pixel value

Padding (border effects)



Convolution: connections and padding

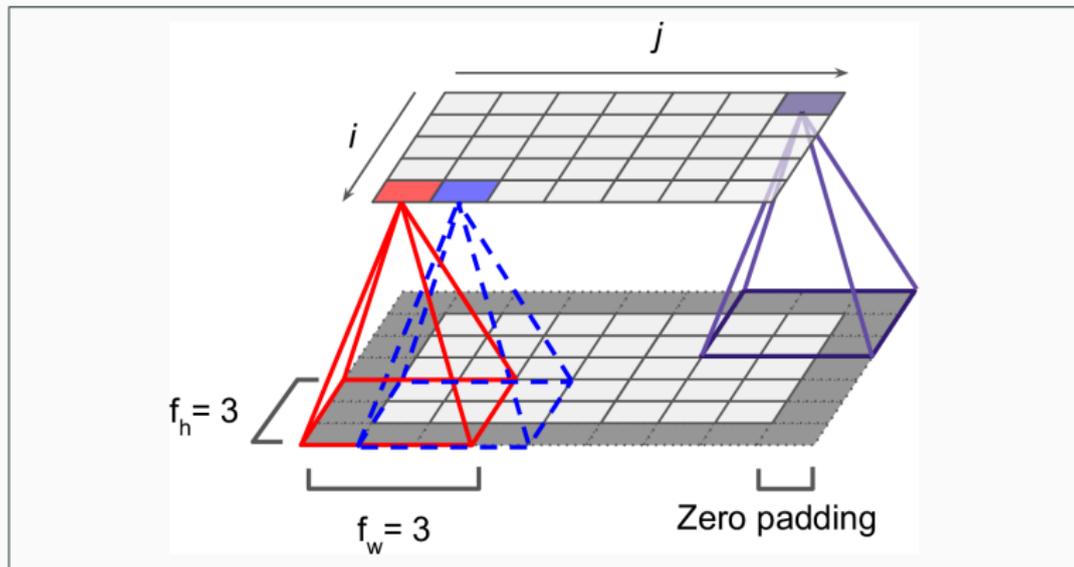


Figure 14-3. Connections between layers and zero padding

Convolution: stride and dimensional reduction

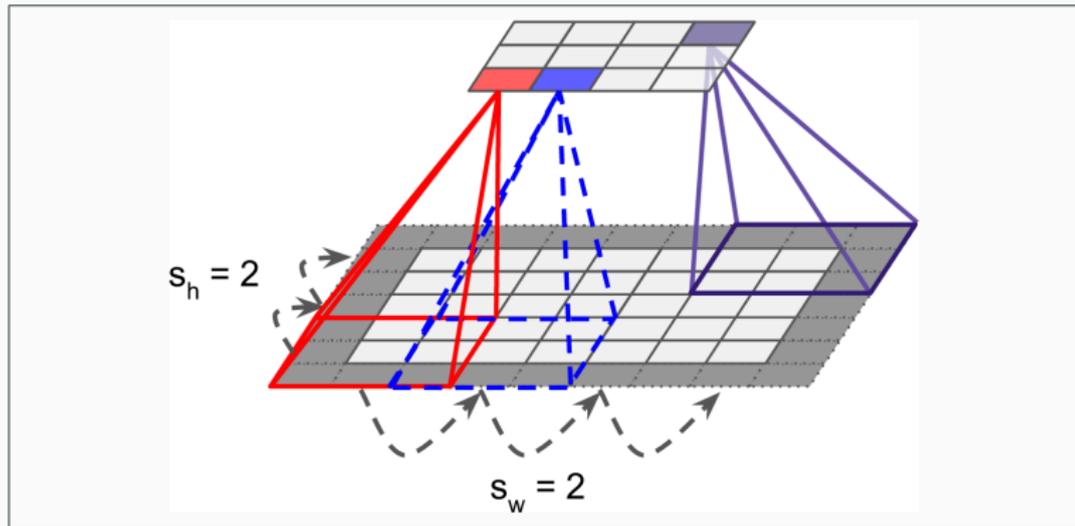


Figure 14-4. Reducing dimensionality using a stride of 2

Convolution: example

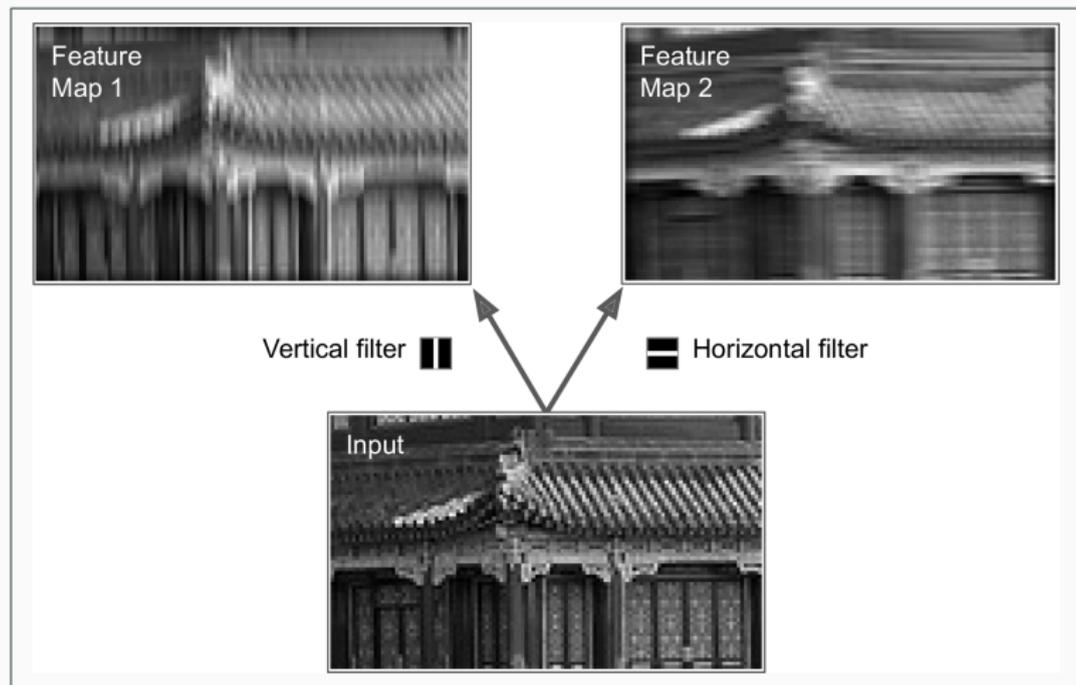
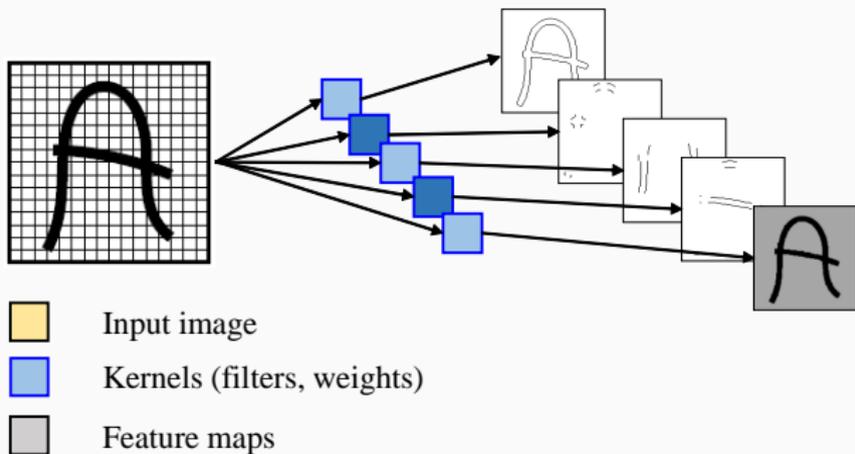


Figure 14-5. Applying two different filters to get two feature maps

Convolution: filter effect

Feature extraction by different kernels



- 컨볼루션은 입력 이미지의 다른 부분들에 존재하는 동일한 특징들을 찾아냄
- Feature map에 있는 뉴런은 template or kernel과 일치되는 경우에 활성화

CNN: convolution layers with multiple feature maps

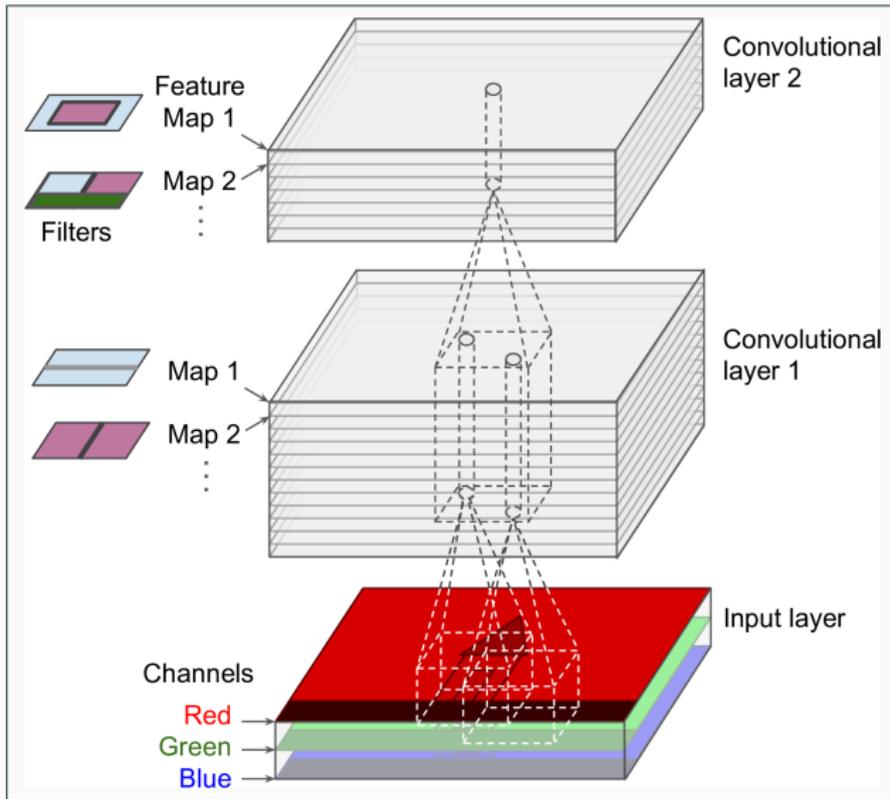
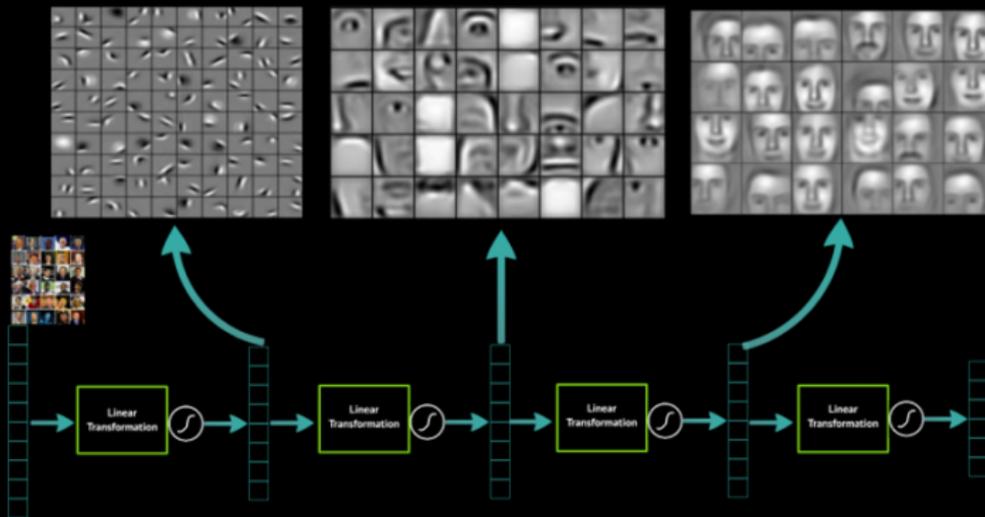


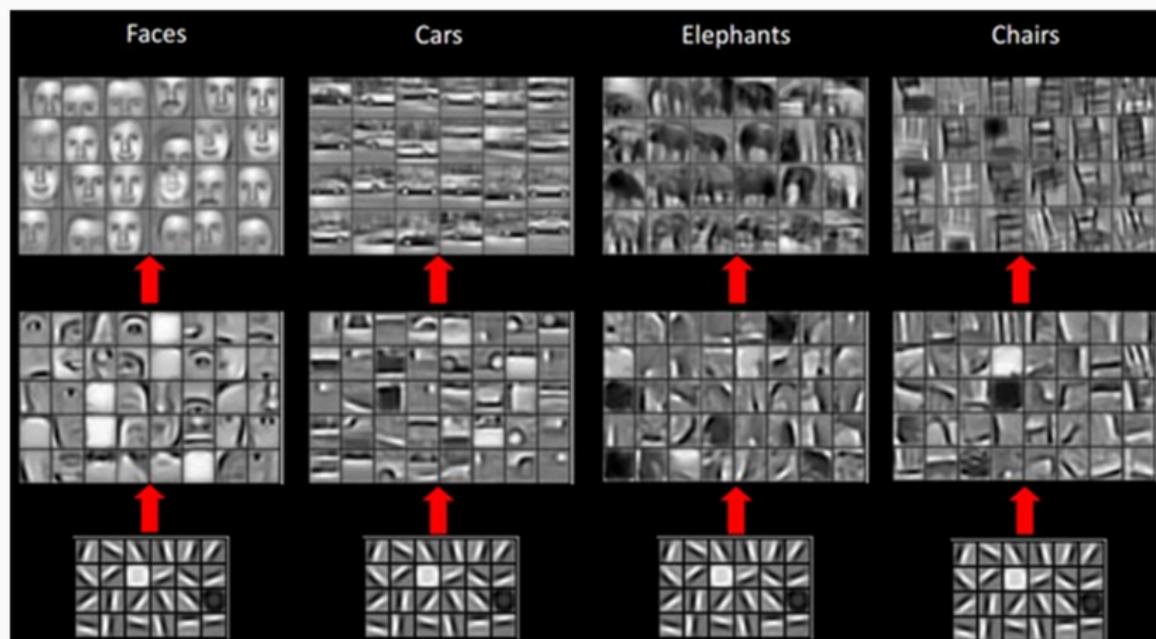
Figure 14-6. Convolution layers with multiple feature maps, and images with three color channels

CNN: hierarchical learning of features

Deep Learning learns layers of features



CNN: hierarchical learning of features



CNN: dimensional reduction by pooling layer

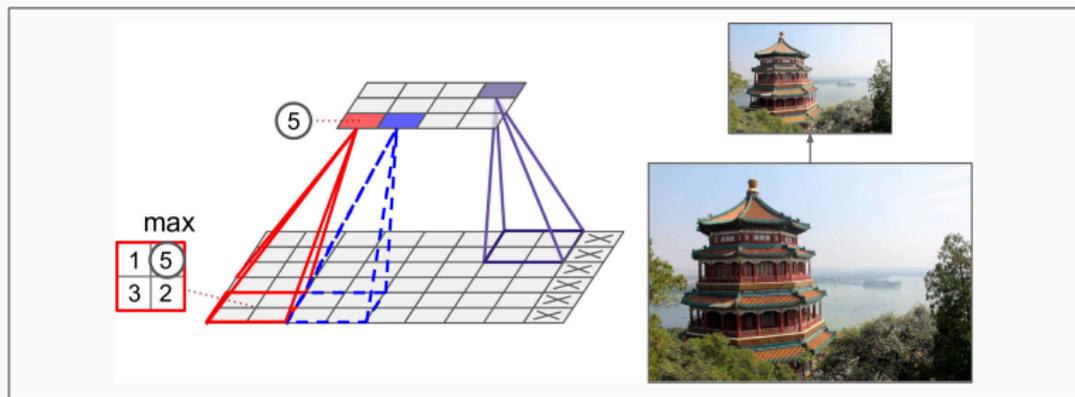


Figure 14-8. Max pooling layer (2×2 pooling kernel, stride 2, no padding)

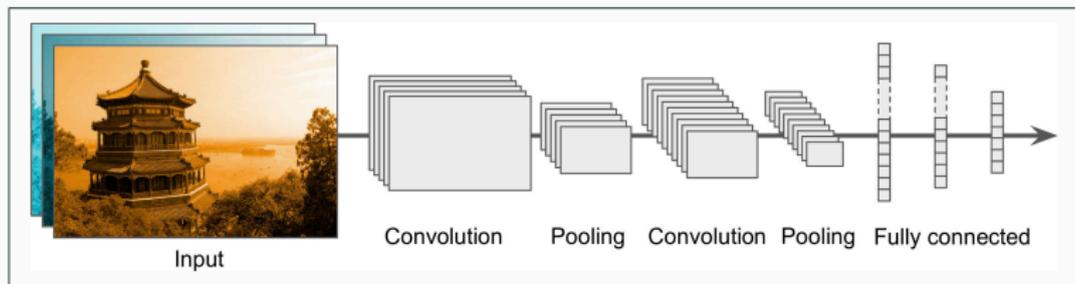


Figure 14-11. Typical CNN architecture

Appendix

Reference and further reading

- “Chap 14 | Deep Computer Vision Using Convolutional Neural Network” of A. Geron, Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow
- “Lecture 10 | Convolutional Neural Networks” of Kwang Il Kim, Machine Learning (2019)