Deep Learning IV: Convolutional Neural Networks

CSci 5525: Machine Learning

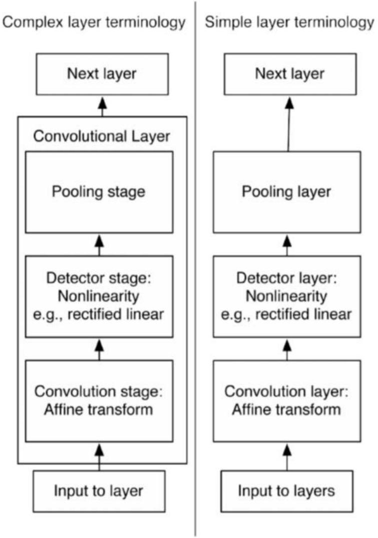
Instructor: Nicholas Johnson

October 29, 2020

Announcements

- HW3 due Nov 10
- No QA session today

Convolutional Network Layer



Convolution Operation

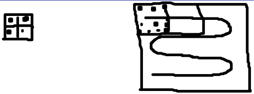
- Convolution of two functions f and g is defined as:

$$s(t) = (f * g)(t) = \int f(a)g(t-a)da$$

- f is input, g is kernel, filter, or receptive field
- In machine learning, often use discrete convolutions

$$(f * g)(t) = \sum_{a=-\infty}^{\infty} f(a)g(t-a)$$

Convolution Operation



- In the case of images we have two-dimensional convolutions:

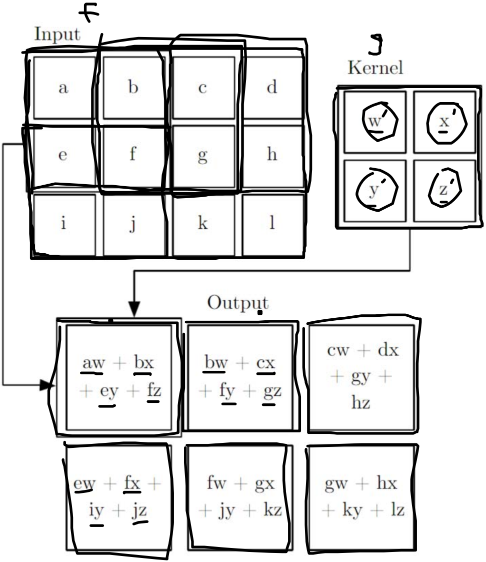
$$(f * g)[i,j] = \sum_m \sum_n \underbrace{f[m,n]}_{\text{image}} \underbrace{g[i-m, j-n]}_{\text{kernel, filter}}$$

$$= \sum_m \sum_n f[i-m, j-n] g[m,n] \quad \text{(commutative property)}$$

→ • In practice, often implemented as cross-correlation

$$s[i,j] = (f * g)[i,j] = \sum_m \sum_n f[i+m, j+n] g[m,n]$$

Example: 2-D convolution



$3\times4 \to \underline{2\times3}$

$n' \times m' \to n \times m$

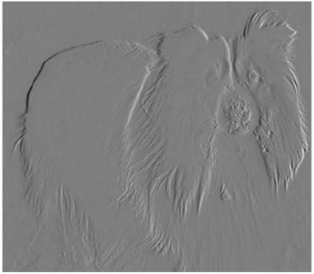
$n' = n + p$

Convolutions as Feature Extraction

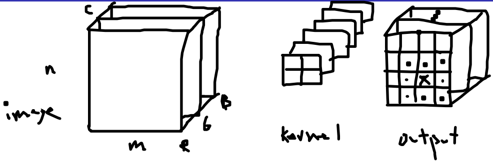
- Different choices of filters lead to different types of feature extractions
  - Edge detection
  - Sharpening
  - Blurring

input

output

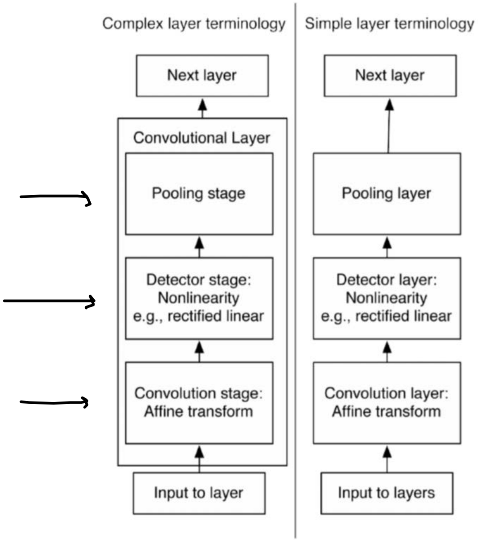


Convolutional Networks: The Three Stages



- Stage 1: Convolution
  - Performs convolutions in parallel
- Stage 2: Detector
  - Nonlinear activation, e.g., using rectified linear unit (ReLU)
- Stage 3: Pooling
  - Update output with summary statistic of nearby outputs

Convolutional Network Layer

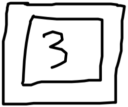


Stage 1: Convolution

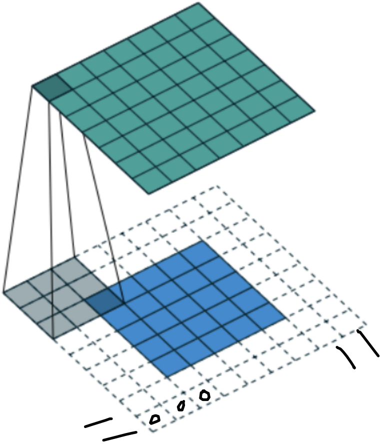
- There are variants on how we apply the convolutional mapping
  - Padding
  - Stride
  - Dilation

Padding

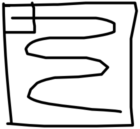
- Padding is when we surround the input matrix with zeros



n×m

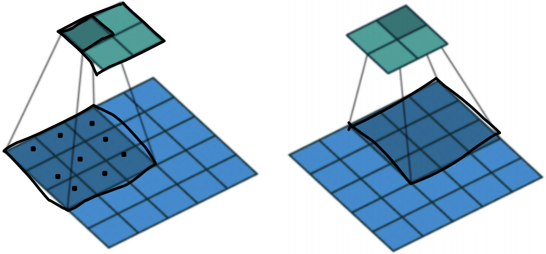


Stride



- Stride is the amount the kernel shifts

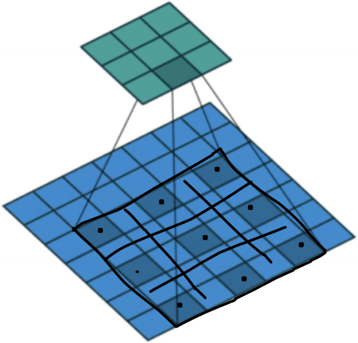
stride = 2



image

Dilation

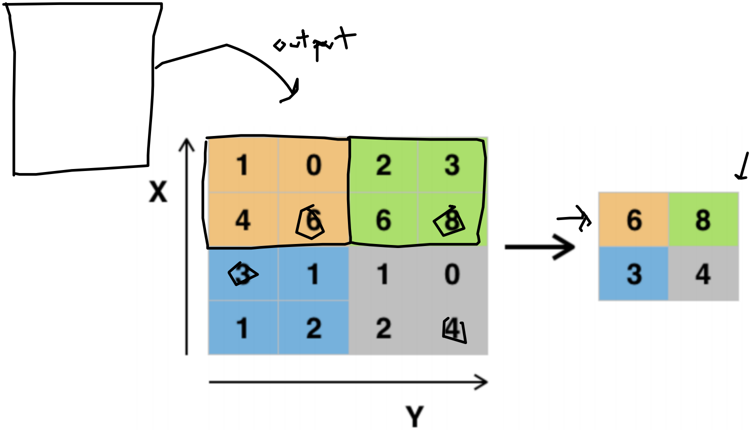
- Convolution applied input with defined gaps, using filter of larger size



Stage 3: Pooling

- Output summary statistic of nearby outputs
- Max-pooling
  - Output maximum of nearby outputs (in a reactangle)
- Other choices: Average, L_2 norm, weighted average
- Pooling leads to invariance
  - Spatial pooling leads to small scale translation invariance
  - Pooling across convolutions leads to corresponding invariance
- Invariance is useful in certain tasks
  - Detects presence of feature rather than its exact location
- Can use fewer pooling units than detector units
  - Reduces output size

Max Pooling



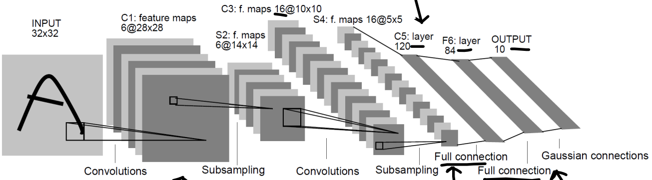
Convolutional Networks: Architecture

LeNet

FF

$w^T x$

$w^T \phi(x)$



feature extraction