Convolutional Neural Networks

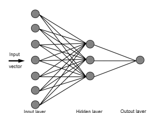
Yuzhen Ye
School of Informatics and Computing, Indiana University

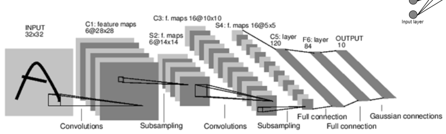
Contents

- CNN basics
- CNN for visual recognition (to explain the concept of convolution)
- CNN for bioinformatics

A Beginner's Guide To Understanding CNN

- Neural Networks with Convolution layers





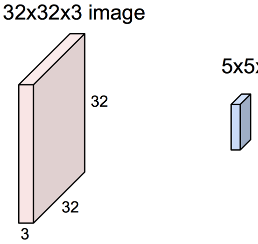
A Full Convolutional Neural Network (LeNet)

Ref 1: http://cs231n.stanford.edu
Ref 2: https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/

Convolution Layer

32x32x3 image

5x5x3 filter



Convolve the filter with the image
i.e. "slide over the image spatially, computing dot products"

Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 7 - 11 27 Jan 2016

Moving Average In 2D

$f[x,y]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$g[x,y]$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

Slide credit: Steve Seitz

5

Correlation filtering

Say the averaging window size is 2k+1 x 2k+1:

$$g(i,j) = \underbrace{\frac{1}{(2k+1)^2}}_{\text{Attribute uniform weight to each pixel}} \underbrace{\sum_{u=-k}^{k} \sum_{v=-k}^{k} f(i+u, j+v)}_{\text{Loop over all pixels in neighborhood around image pixel } f[i,j]}$$

Now generalize to allow different weights depending on neighboring pixel's relative position:

$$g(i,j) = \sum_{u=-k}^{k} \sum_{v=-k}^{k} \underbrace{h(u,v)}_{\text{Non-uniform weights}} f(i+u, j+v)$$

Slide adapted from Kristen Grauman

6

Correlation filtering

$$g(i,j) = \sum_{u=-k}^{k} \sum_{v=-k}^{k} h(u,v) f(i+u, j+v)$$

This is called cross-correlation, denoted $g = h \otimes f$

Filtering an image: replace each pixel with a linear combination of its neighbors.

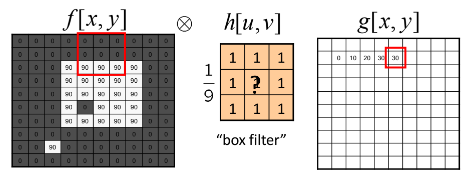
The filter "kernel" or "mask" *h*[*u,v*] is the prescription for the weights in the linear combination.

Slide credit: Michael S. Ryoo

7

Averaging filter

- What values belong in the kernel *h* for the moving average example?



"box filter"

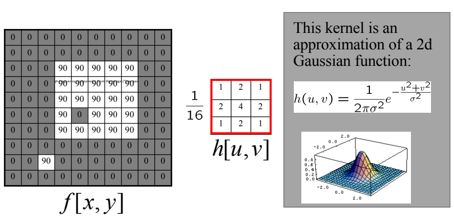
$$g = h \otimes f$$

Slide credit: Michael S. Ryoo

8

Gaussian filter

- What if we want nearest neighboring pixels to have the most influence on the output?



$$h(u,v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{\sigma^2}}$$

Slide credit: Steve Seitz

9

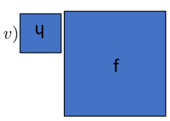
Convolution

- Convolution is a simple mathematical operation which is fundamental to many common image processing operators.
- Convolution is performed by sliding the kernel over the image, generally starting at the top left corner, so as to move the kernel through all the positions where the kernel fits entirely within the boundaries of the image.
- Convolution:
  - Flip the filter in both dimensions (bottom to top, right to left)
  - Then apply cross-correlation

$$g(i,j) = \sum_{u=-k}^{k} \sum_{v=-k}^{k} h(u,v) f(i-u, j-v)$$

$$g = h * f$$

Notation for convolution operator



Slide credit: Michael S. Ryoo

10

Convolution vs. correlation

Convolution

$$g(i,j) = \sum_{u=-k}^{k} \sum_{v=-k}^{k} h(u,v) f(i-u, j-v)$$

$$g = h * f$$

Cross-correlation

$$g(i,j) = \sum_{u=-k}^{k} \sum_{v=-k}^{k} h(u,v) f(i+u, j+v)$$

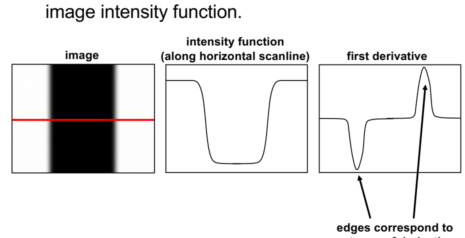
$$g = h \otimes f$$

Slide adapted from Kristen Grauman

11

Derivatives and edges

An edge is a place of rapid change in the image intensity function.



Slide credit: Svetlana Lazebnik

12

Derivatives with convolution

For 2D function, f(x,y), the partial derivative is:

$$\frac{\partial f(x,y)}{\partial x} = \lim_{\varepsilon \to 0} \frac{f(x+\varepsilon, y) - f(x,y)}{\varepsilon}$$

For discrete data, we can approximate using finite differences:

$$\frac{\partial f(x,y)}{\partial x} \approx \frac{f(x+1, y) - f(x,y)}{1}$$

To implement above as convolution, what would be the associated filter?

Slide credit: Kristen Grauman

13

Partial derivatives of an image

$\frac{\partial f(x,y)}{\partial x}$ $\frac{\partial f(x,y)}{\partial y}$

-1	1

	? or	
-1		1
1		-1

Which shows changes with respect to x?

Slide credit: Kristen Grauman (showing filters for correlation)

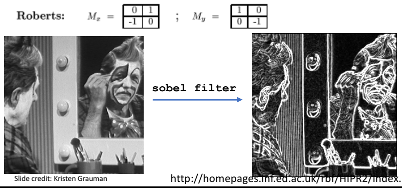
14

Filters as feature (edge) detectors

Prewitt: $M_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$; $M_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$

Sobel: $M_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$; $M_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$

Roberts: $M_x = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$; $M_y = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$



Slide credit: Kristen Grauman http://homepages.inf.ed.ac.uk/rbf/HIPR2/index.htm

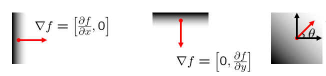
15

Image gradient

The gradient of an image:

$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$

The gradient points in the direction of most rapid change in intensity



$\nabla f = \left[\frac{\partial f}{\partial x}, 0\right]$ $\nabla f = \left[0, \frac{\partial f}{\partial y}\right]$ $\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$

The **gradient direction** (orientation of edge normal) is given by:

$$\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$$

The **edge strength** is given by the gradient magnitude

$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

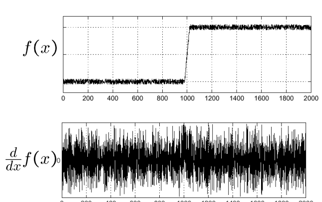
Slide credit: Steve Seitz

16

Effects of noise

Consider a single row or column of the image

- Plotting intensity as a function of position gives a signal



Where is the edge?

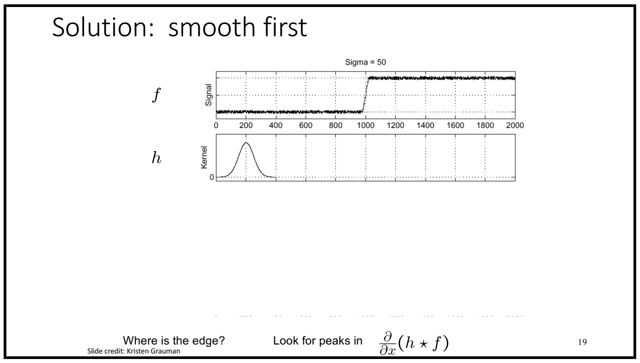
Slide credit: Steve Seitz

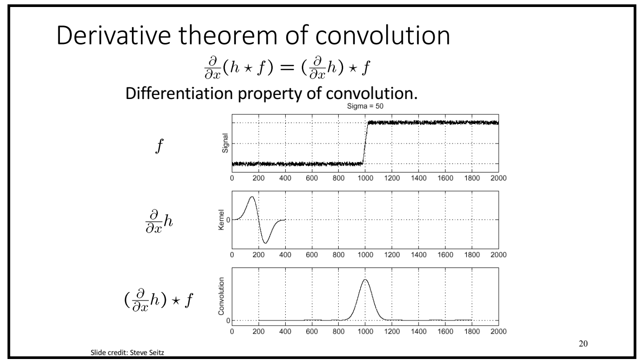
17

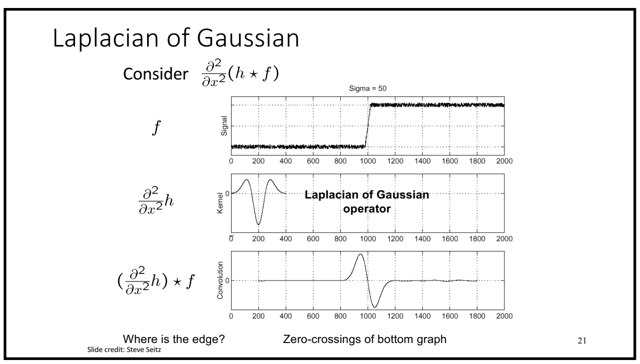
Effects of noise

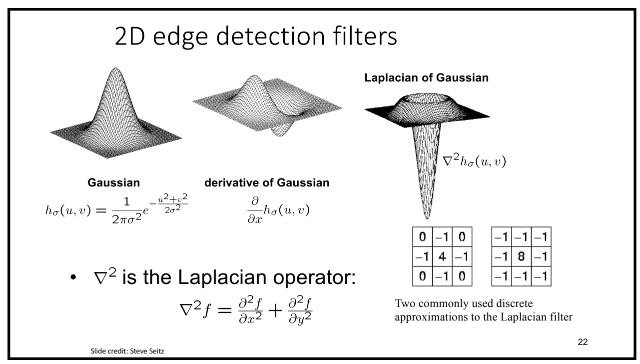
- Difference filters respond strongly to noise
  - Image noise results in pixels that look very different from their neighbors
  - Generally, the larger the noise the stronger the response
- What can we do about it?

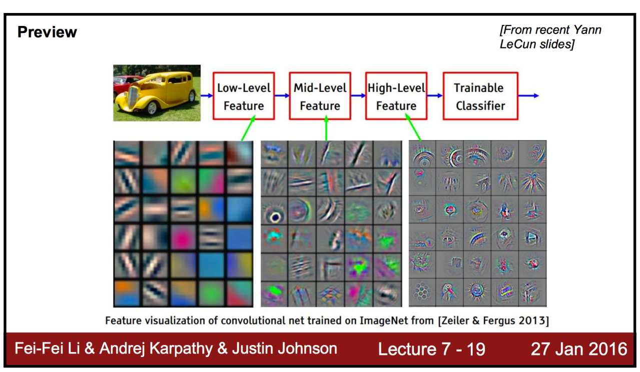
Slide credit: Michael S. Ryoo

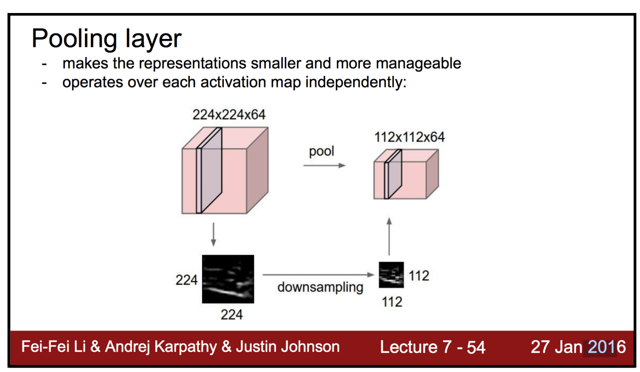

Solution: smooth first
Sigma = 50
f
Signal
h
Kernel
Where is the edge?
Look for peaks in $\frac{\partial}{\partial x}(h \star f)$
Slide credit: Kristen Grauman
19


Derivative theorem of convolution
$\frac{\partial}{\partial x}(h \star f) = (\frac{\partial}{\partial x}h) \star f$
Differentiation property of convolution.
Sigma = 50
f
$\frac{\partial}{\partial x}h$
$(\frac{\partial}{\partial x}h) \star f$
Signal
Kernel
Convolution
Slide credit: Steve Seitz
20

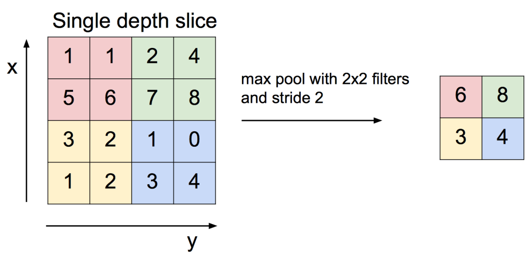

Laplacian of Gaussian
Consider $\frac{\partial^2}{\partial x^2}(h \star f)$
Sigma = 50
f
$\frac{\partial^2}{\partial x^2}h$
Laplacian of Gaussian operator
$(\frac{\partial^2}{\partial x^2}h) \star f$
Signal
Kernel
Convolution
Where is the edge?
Zero-crossings of bottom graph
Slide credit: Steve Seitz
21


2D edge detection filters
Laplacian of Gaussian
Gaussian
derivative of Gaussian
$h_\sigma(u,v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$
$\frac{\partial}{\partial x} h_\sigma(u,v)$
$\nabla^2 h_\sigma(u,v)$
• ∇^2 is the Laplacian operator:
$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$
0 −1 0
−1 4 −1
0 −1 0
−1 −1 −1
−1 8 −1
−1 −1 −1
Two commonly used discrete approximations to the Laplacian filter
Slide credit: Steve Seitz
22


Preview
[From recent Yann LeCun slides]
Low-Level Feature
Mid-Level Feature
High-Level Feature
Trainable Classifier
Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]
Fei-Fei Li & Andrej Karpathy & Justin Johnson
Lecture 7 - 19
27 Jan 2016

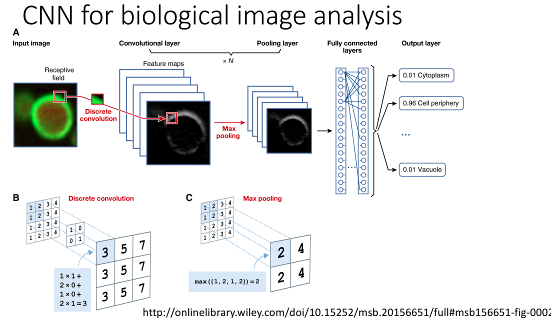

Pooling layer
- makes the representations smaller and more manageable
- operates over each activation map independently:
224x224x64
pool
112x112x64
224
224
downsampling
112
112
Fei-Fei Li & Andrej Karpathy & Justin Johnson
Lecture 7 - 54
27 Jan 2016

MAX POOLING



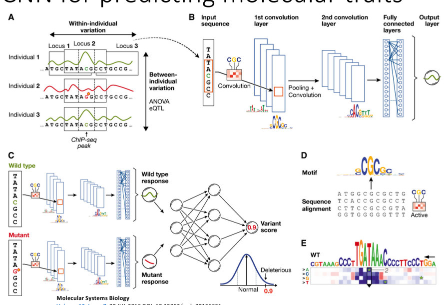
Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 7 - 55 27 Jan 2016

CNN for biological image analysis



http://onlinelibrary.wiley.com/doi/10.15252/msb.20156651/full#msb156651-fig-0002

CNN for predicting molecular traits



Molecular Systems Biology
Volume 12, Issue 7, 29 JUL 2016 DOI: 10.15252/msb.20156651
http://onlinelibrary.wiley.com/doi/10.15252/msb.20156651/full#msb156651-fig-0002

Input data: one-dimensional genomic sequences with one channel per nucleotide

[Visual recognition: 2D-image with three color channels]