CPSC 340: Machine Learning and Data Mining

Convolutional Neural Networks

Fall 2016

Admin

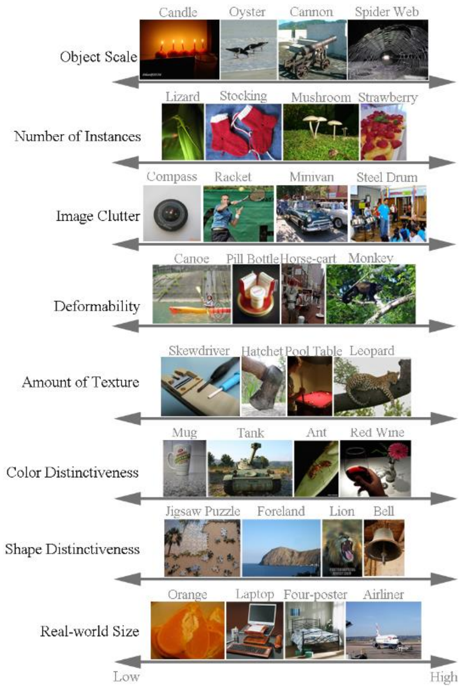
- Assignment 5:
  - Due Friday, 1 late day to hand in Monday, etc.
- Assignment 6:
  - Due next Friday (usual late day policy, assuming phantom “classes”).
- Final:
  - December 12 (8:30am – HEBB 100)
  - Covers Assignments 1-6.
  - Final from last year and list of topics will be posted.
  - Closed-book, cheat sheet: 4-pages each double-sided.

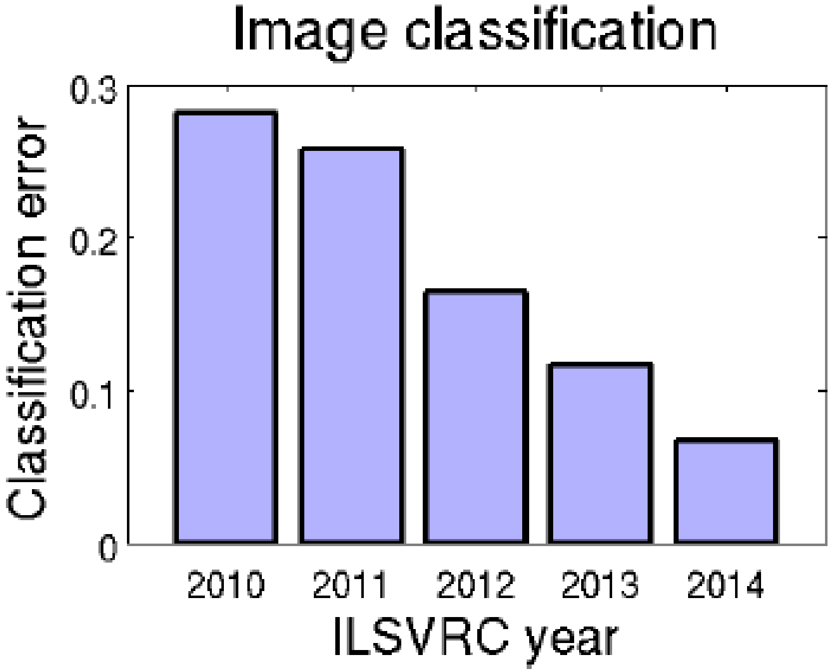
Last Lectures: Deep Learning

- We've been discussing neural network / deep learning models:

$$y_i = w^T h(W^{(m)} h(W^{(m-1)} h(\cdots\cdots W^{(2)} h(W^{(1)} x_i))\cdots))$$

- On Friday we discussed unprecedented vision/speech performance.





https://arxiv.org/pdf/1409.0575v3.pdf

Last Lectures: Deep Learning

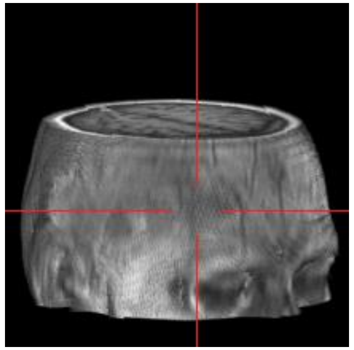
- We've been discussing neural network / deep learning models:

$$y_i = w^T h(W^{(m)} h(W^{(m-1)} h(\cdots\cdots W^{(2)} h(W^{(1)} x_i))\cdots))$$

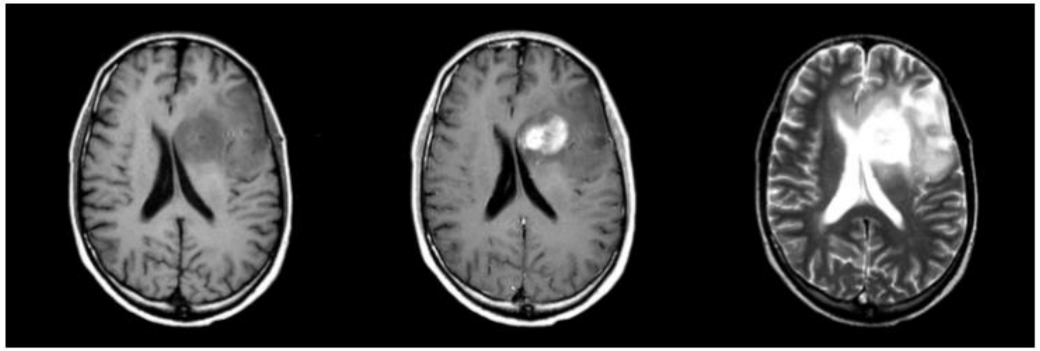
- On Monday we discussed heuristics to make it work:
  - Parameter initialization and data transformations.
  - Setting the step size(s) in stochastic gradient.
  - Alternative non-linear functions like ReLU.
  - Different forms of regularization:
    - L2-regularization, early stopping, dropout.
- These are often still not enough to get deep models working.

Motivation: Automatic Brain Tumor Segmentation

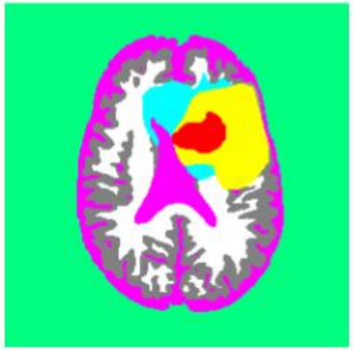
- Task: segmentation tumors and normal tissue in multi-modal MRI data.



Input:



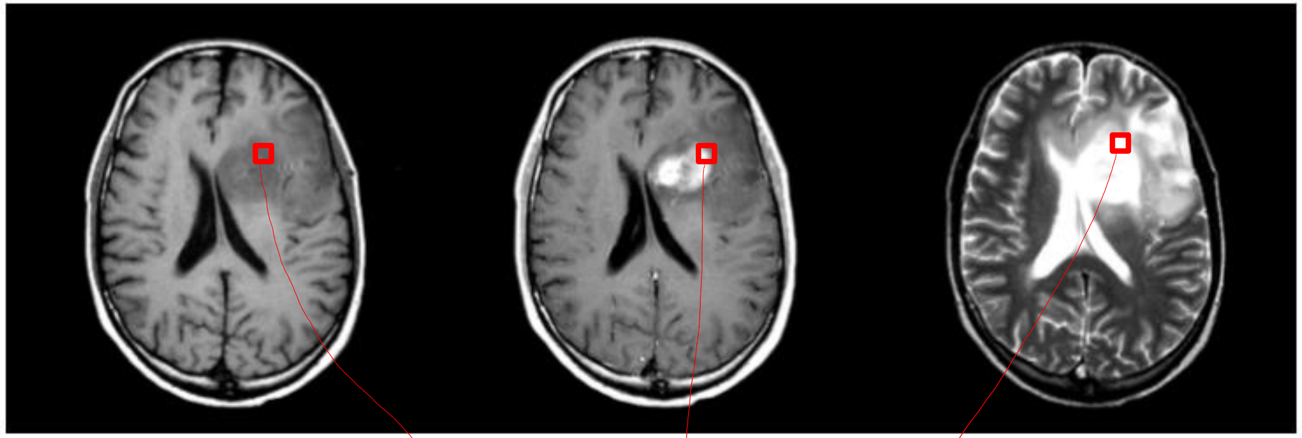
Output:



- Applications:
  - Radiation therapy target planning, quantifying treatment responses.
  - Mining growth patterns, image-guided surgery.
- Challenges:
  - Variety of tumor appearances, similarity to normal tissue.
  - “You are never going to solve this problem.”

Naïve Voxel-Level Classifier

- We could treat classifying a voxel as supervised learning:

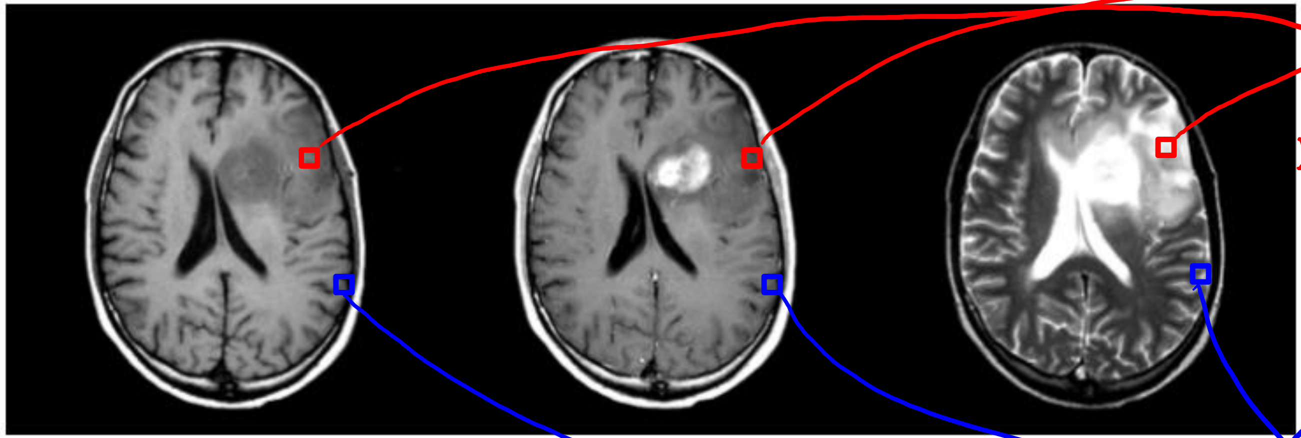


$x_i = (98, 187, 246)$ $\qquad$ $y_i = \text{"tumour"}$

- We can formulate predicting y_i given x_i as supervised learning.
- But it doesn't work at all with these features.

Need for Context

- The individual voxel values are almost meaningless:
  - This x_i could lead to different y_i.



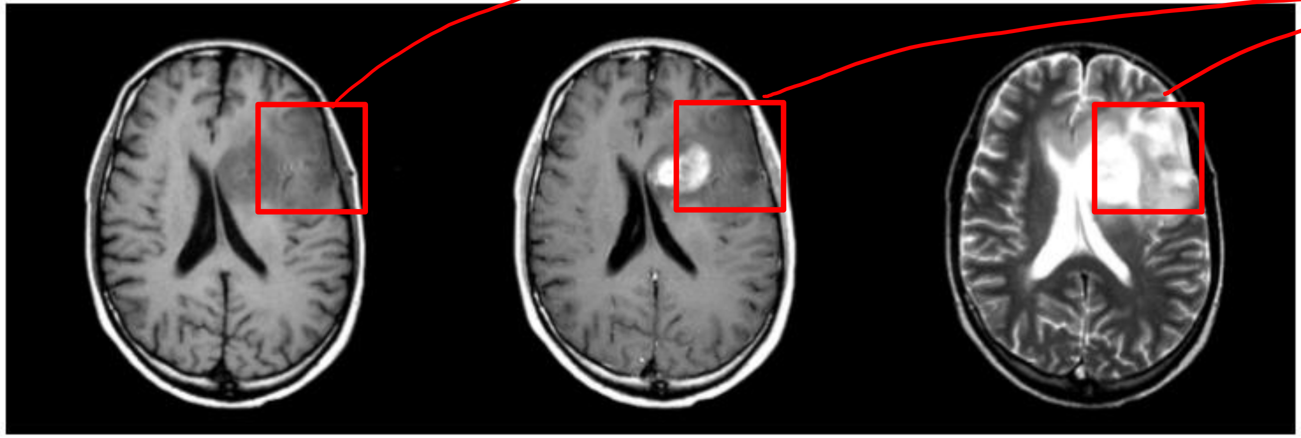
$x_i = (87, 54, 173)$ $y_i =$ "tumour"

$x_i = (87, 54, 173)$ $y_i =$ "normal"

- Intensities not standardized.
- Non-trivial overlap in signal for different tissue types.
- "Partial volume" effects at boundaries of tissue types.

Need for Context

- We need to represent the spatial "context" of the voxel.



$x_i = (\overbrace{- - - - -}, \overbrace{- - - - -}, \overbrace{- - - - -})$

  – Include all the values of neighbouring voxels?
    - Using all voxels requires lots of data to find patterns.
  – Measure summary statistics (mean, variance, etc.) of the neighbourhood?
    - Loses spatial information present in voxels.
  – Standard approach is uses convolutions to represent neighbourhood.

1D Convolution

- 1D convolution input:
  - Signal 'x' which is a vector length 'n'.
    - Indexed by i=1,2,…,n.
  - Filter 'w' which is a vector of length '2m+1':
    - Indexed by i=-m,-m+1,…-2,0,1,2,…,m-1,m

$$x = [0 \quad 1 \quad 1 \quad 2 \quad 3 \quad 5 \quad 8 \quad 13]$$

$$w = [\underset{w_{-2}}{0} \quad \underset{w_{-1}}{-1} \quad \underset{w_0}{2} \quad \underset{w_1}{-1} \quad \underset{w_2}{0}]$$

- 1D convolution output:
  - New vector 'z' of length 'n' with elements:

$$z_i = w_{-m}x_{i-m} + w_{-m+1}x_{i-m+1} + \cdots + w_{m+1}x_{i+m+1} + w_m x_{i+m}$$

$$z_4 = 0\cdot 1 + (-1)\cdot 1 + 2\cdot 2 + (-1)\cdot 3 + 0\cdot 5$$
$$= 0 - 1 + 4 - 3 + 0 = 0$$

1D Convolution Examples

- Element z_i of 1D convolution is given by:

$$z_i = w_{-m} x_{i-m} + w_{-m+1} x_{i-m+1} + \cdots + w_{m+1} x_{i+m+1} + w_m x_{i+m}$$

- Examples:

  Let $x = [0 \;\; 1 \;\; 1 \;\; 2 \;\; 3 \;\; 5 \;\; 8 \;\; 13]$

  - "Identity"

    $w = [0 \;\; 1 \;\; 0]$

    $z = [0 \;\; 1 \;\; 1 \;\; 2 \;\; 3 \;\; 5 \;\; 8 \;\; 13]$

    $0 \cdot x_0 + 1 \cdot x_1 + 0 \cdot x_2$ $\quad$ $0 \cdot x_1 + 1 \cdot x_2 + 0 \cdot x_3$

  - "Translation"

    $w = [0 \;\; 0 \;\; 1]$

    $z = [1 \;\; 1 \;\; 2 \;\; 3 \;\; 5 \;\; 8 \;\; 13 \;\; ?]$

    $0 \cdot x_0 + 0 \cdot x_1 + 1 \cdot x_2$

1D Convolution Examples

- Element z_i of 1D convolution is given by:

$$z_i = w_{-m} x_{i-m} + w_{-m+1} x_{i-m+1} + \cdots + w_{m+1} x_{i+m+1} + w_m x_{i+m}$$

- Examples:

Let $x = [0 \;\; 1 \;\; 1 \;\; 2 \;\; 3 \;\; 5 \;\; 8 \;\; 13]$

  - "Identity"

$w = [0 \;\; 1 \;\; 0]$

$z = [0 \;\; 1 \;\; 1 \;\; 2 \;\; 3 \;\; 5 \;\; 8 \;\; 13]$

$\underbrace{0 \;\; 1 \;\; 1}_{\text{average}} \qquad \underbrace{2 \;\; 3 \;\; 5}_{\text{average}}$

  - "Average"

$w = [\tfrac{1}{3} \;\; \tfrac{1}{3} \;\; \tfrac{1}{3}]$

$z = [? \;\; \tfrac{2}{3} \;\; 1\tfrac{1}{3} \;\; 2 \;\; 3\tfrac{1}{3} \;\; 5\tfrac{1}{3} \;\; 8\tfrac{2}{3} \;\; ?]$

Boundary Issue

- What can we about the "?" at the edges?

If $x = [0\ 1\ 1\ 2\ 3\ 5\ 8\ 13]$ and $w = [\tfrac{1}{3}\ \tfrac{1}{3}\ \tfrac{1}{3}]$ then $z = [?\ \tfrac{2}{3}\ 1\tfrac{1}{3}\ 2\ 3\tfrac{1}{3}\ 5\tfrac{1}{3}\ 8\tfrac{2}{3}\ ?]$

- Can assign values past the boundaries:
  - "Zero": $x = 0\ 0\ 0\ [0\ 1\ 1\ 2\ 3\ 5\ 8\ 13]\ 0\ 0\ 0$
  - "Replicate": $x = 0\ 0\ 0\ [0\ 1\ 1\ 2\ 3\ 5\ 8\ 13]\ 13\ 13\ 13$
  - "Mirror": $x = 2\ 1\ 1\ [0\ 1\ 1\ 2\ 3\ 5\ 8\ 13]\ 8\ 5\ 3$
- Or just ignore the "?" values and return a shorter vector:

$$z = [\tfrac{2}{3}\ 1\tfrac{1}{3}\ 2\ 3\tfrac{1}{3}\ 5\tfrac{1}{3}\ 8\tfrac{2}{3}]$$

1D Convolution in Matrix Notation

- Each element of a convolution is an inner product:

$$z_i = w_{-m}x_{i-m} + w_{-m+1}x_{i-m+1} + \cdots + w_{m+1}x_{i+m+1} + w_m x_{i+m}$$

$$= \sum_{j=-m}^{m} w_j x_{i+j}$$

$$= w^T x_{(i-m:i+m)}$$

$$= \tilde{w}^T x \quad \text{where } \tilde{w} = [0 \;\; 0 \;\; 0 \;\; \underbrace{\text{———} \; w \; \text{———}}_{\text{positions } i-m \text{ through } i+m} \;\; 0 \;\; 0]$$

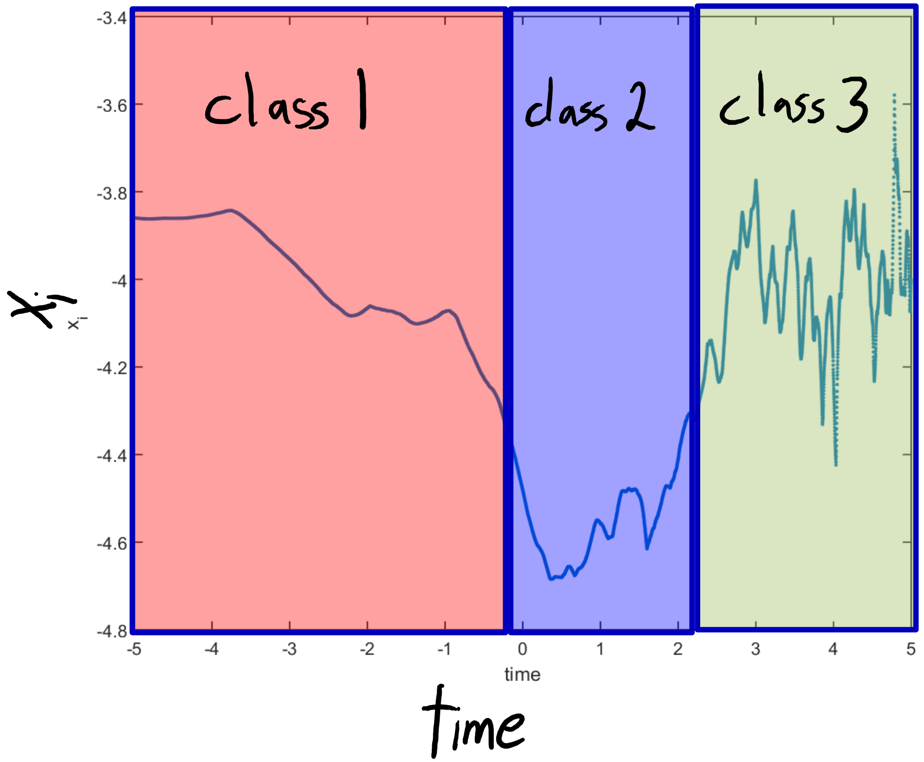
- So convolution is a matrix multiplication:

$$z = \tilde{W}x \quad \text{where } \tilde{W} = \begin{bmatrix} \text{———} & w & \text{———} & 0 & 0 & 0 \\ 0 & \text{———} & w & \text{———} & 0 & 0 \\ 0 & 0 & \text{———} & w & \text{———} & 0 \\ 0 & 0 & 0 & \text{———} & w & \text{———} \end{bmatrix}$$

matrix can be very sparse and only has 2m+1 variables.

Why is this useful?

- Consider a 1D dataset:
  - Want to classify each time into y_i in {1,2,3}.



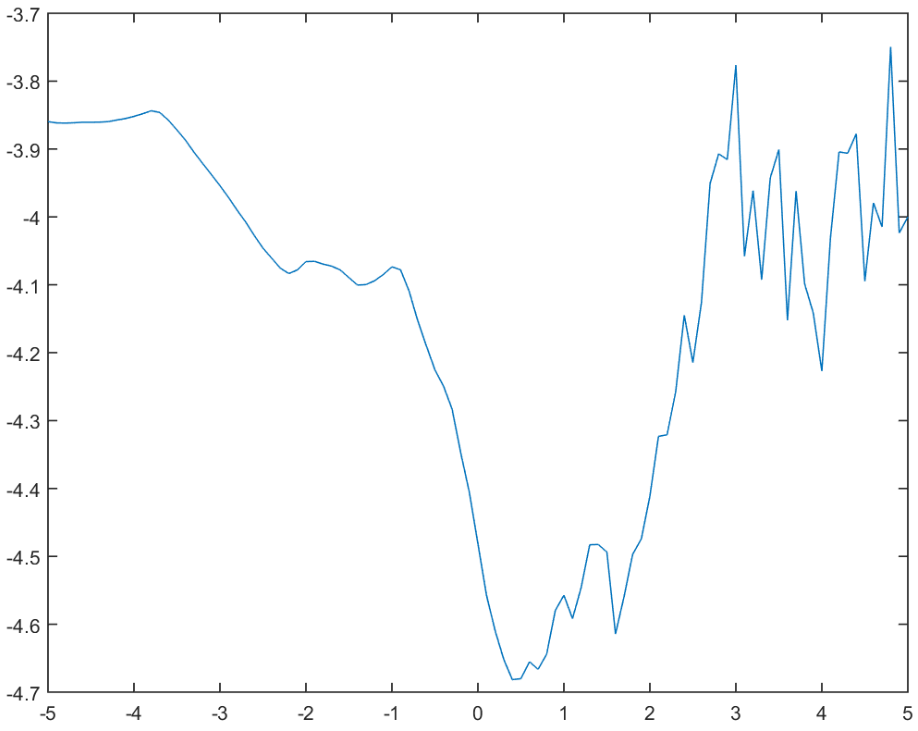
  - Example: sound data.

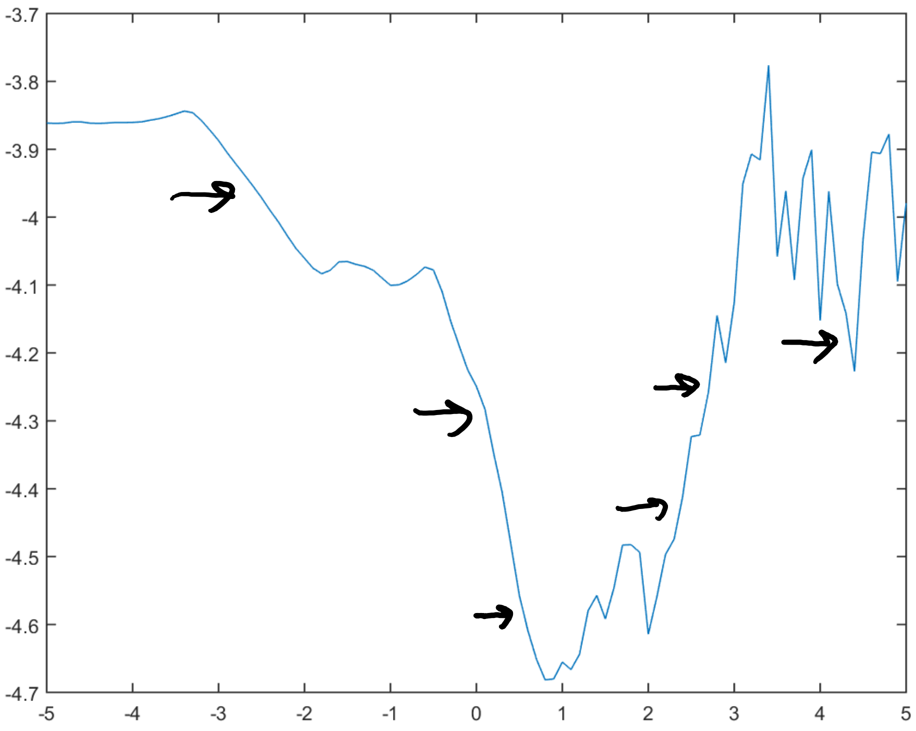
- Easy to distinguish class 2 from the other classes (x_i are smaller).
- Harder to distinguish between class 1 and class 3 (similar x_i range).
  - But convolutions can represent that class 3 is more "spiky".

1D Convolution Examples

- Translation convolution shift signal:

$w = [1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$

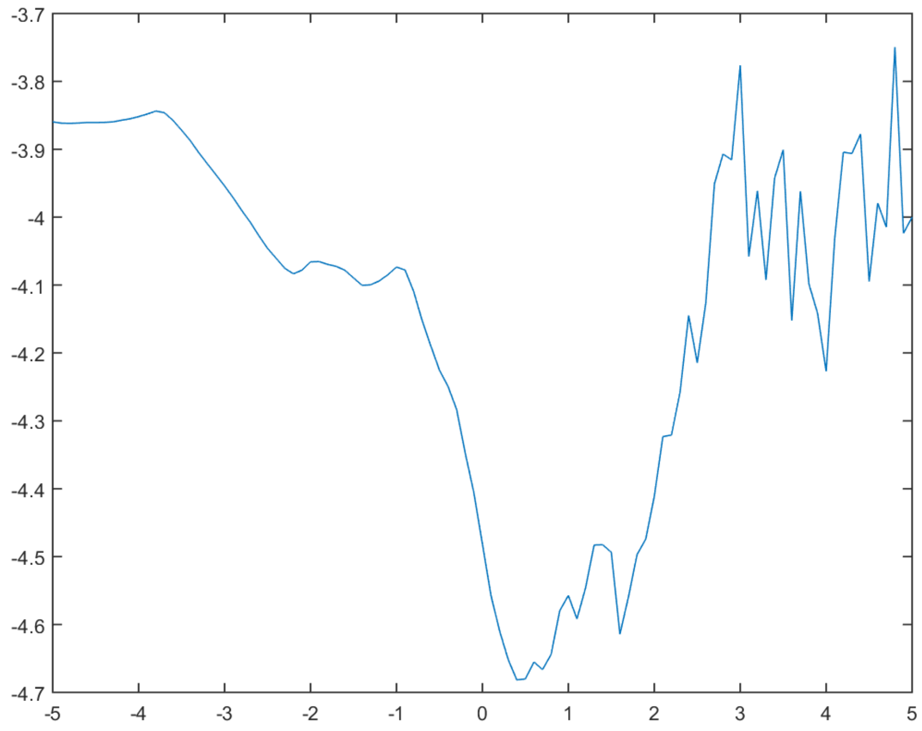


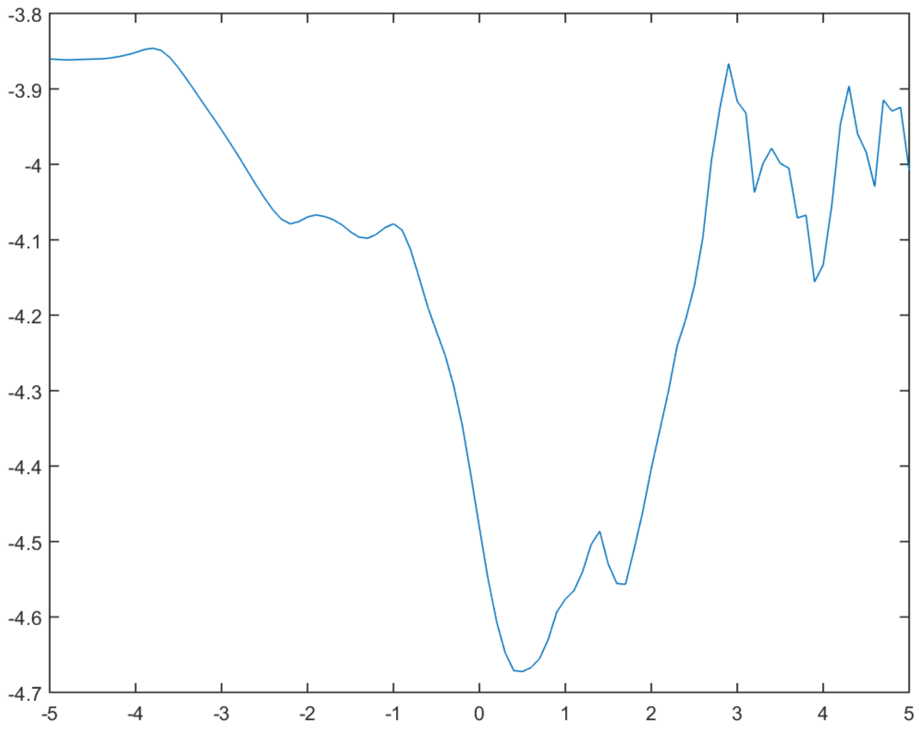


1D Convolution Examples

- Averaging convolution computes local mean:

$$w = [\tfrac{1}{3} \quad \tfrac{1}{3} \quad \tfrac{1}{3}]$$

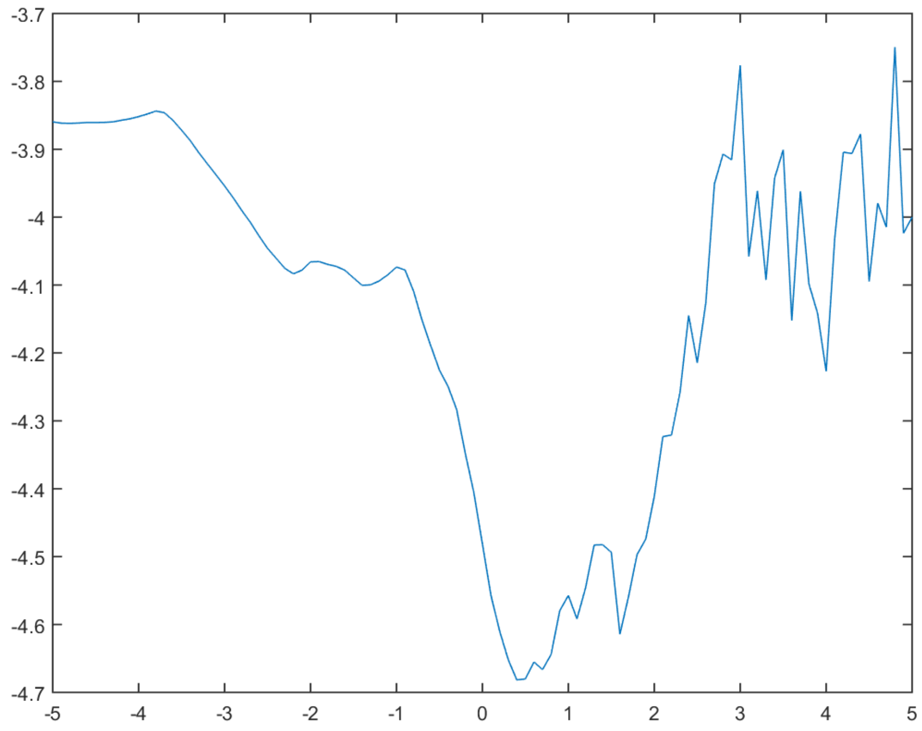


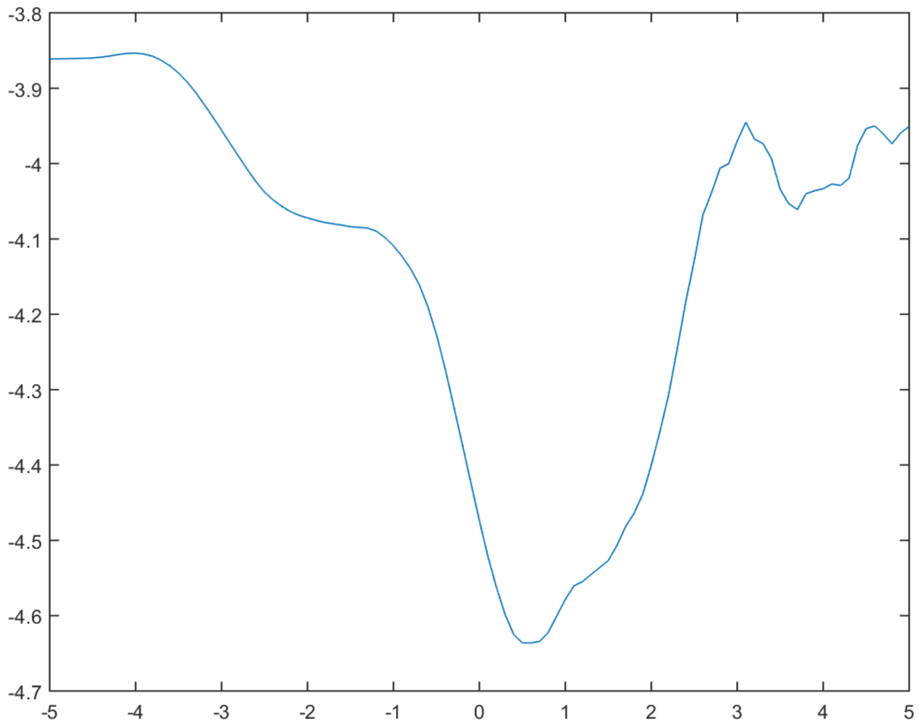


1D Convolution Examples

- Averaging over bigger window gives coarser view of signal:

$$w = [\tfrac{1}{9}\ \tfrac{1}{9}\ \tfrac{1}{9}\ \tfrac{1}{9}\ \tfrac{1}{9}\ \tfrac{1}{9}\ \tfrac{1}{9}\ \tfrac{1}{9}\ \tfrac{1}{9}]$$



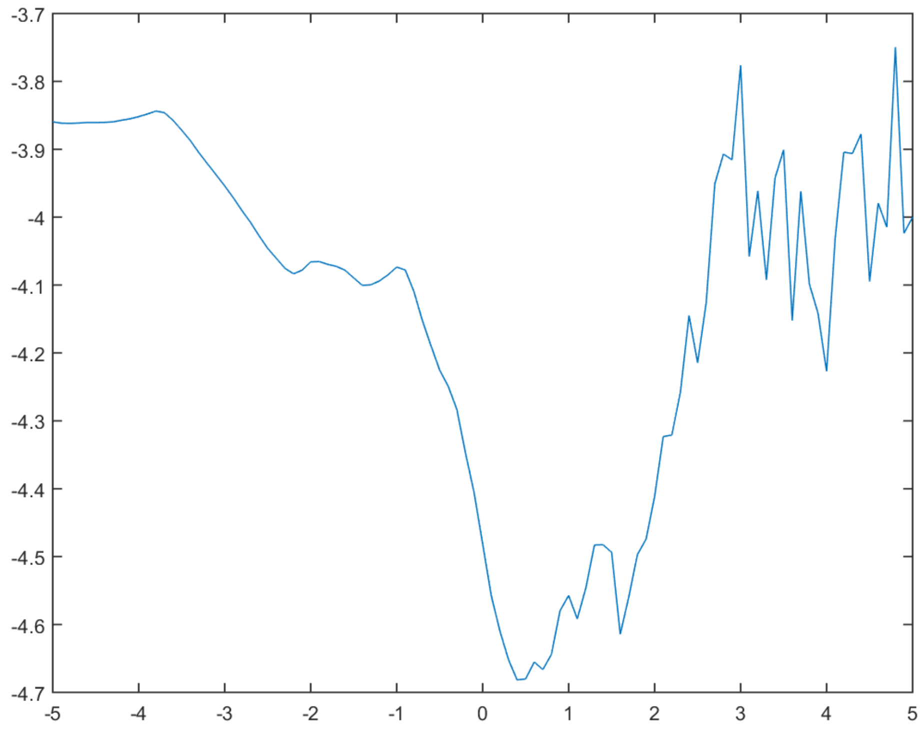


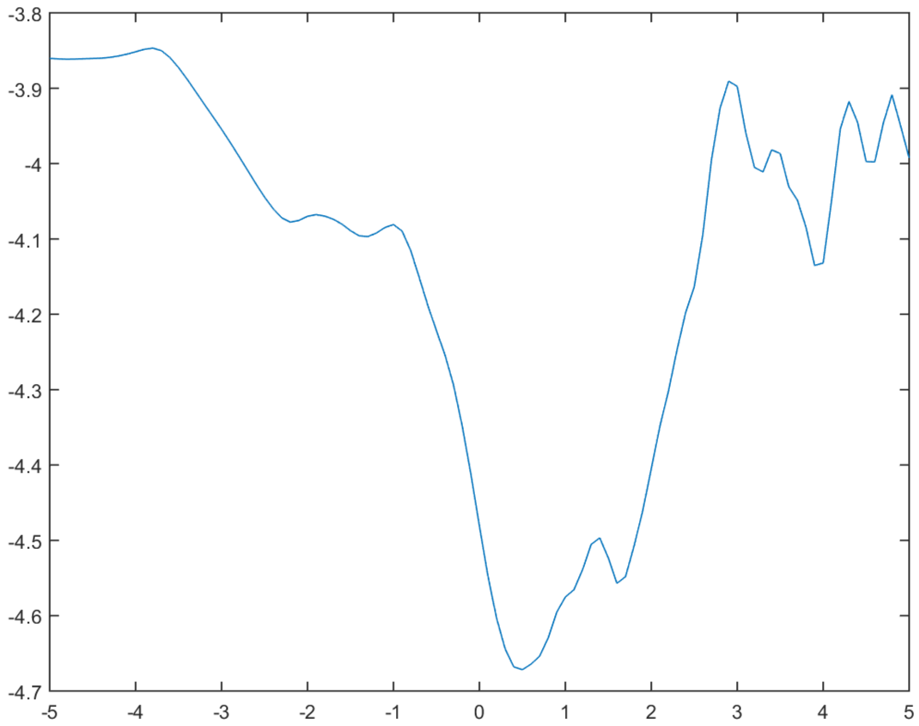
1D Convolution Examples

- Gaussian convolution blurs signal: $w_i \propto \exp\left(-\frac{i^2}{2\sigma^2}\right)$
  - Compared to averaging it's more smooth and maintains peaks better.

$W = [0.0001 \quad 0.0044 \quad 0.0540 \quad 0.2420 \quad 0.3989 \quad 0.2420 \quad 0.0540 \quad 0.0044 \quad 0.0001]$

$(\sigma = 1, \; m = 4)$

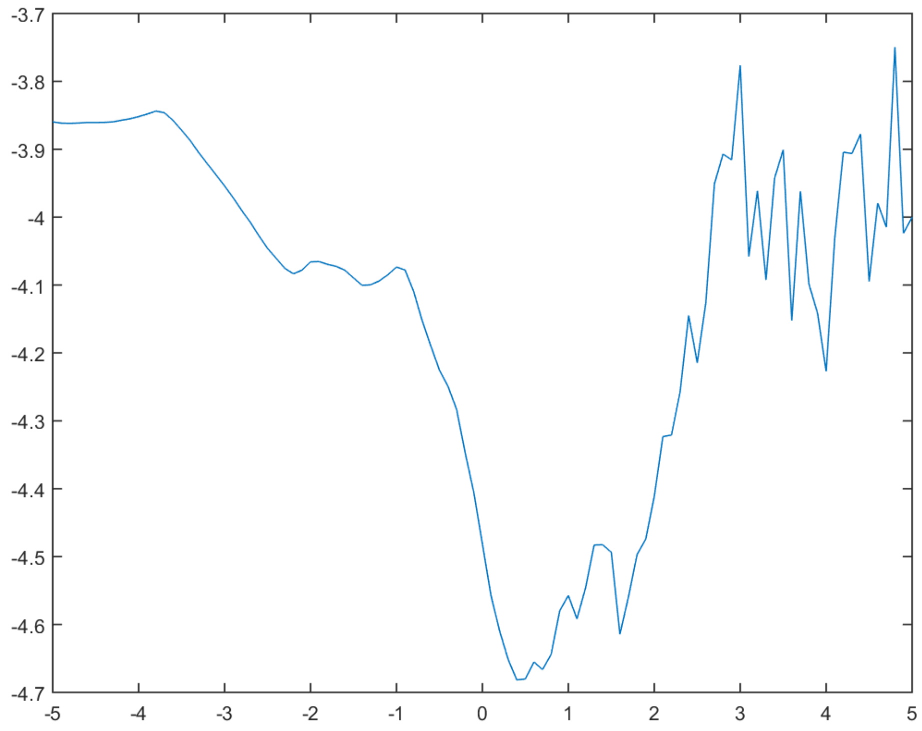


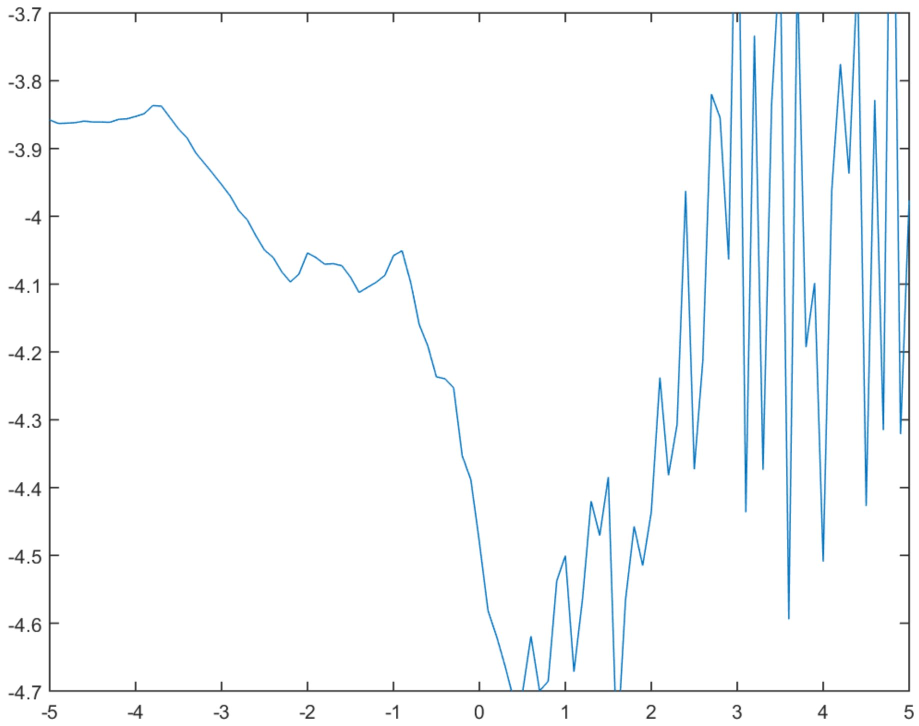


1D Convolution Examples

- Sharpen convolution enhances peaks.

$$w = [-1 \quad 3 \quad -1]$$

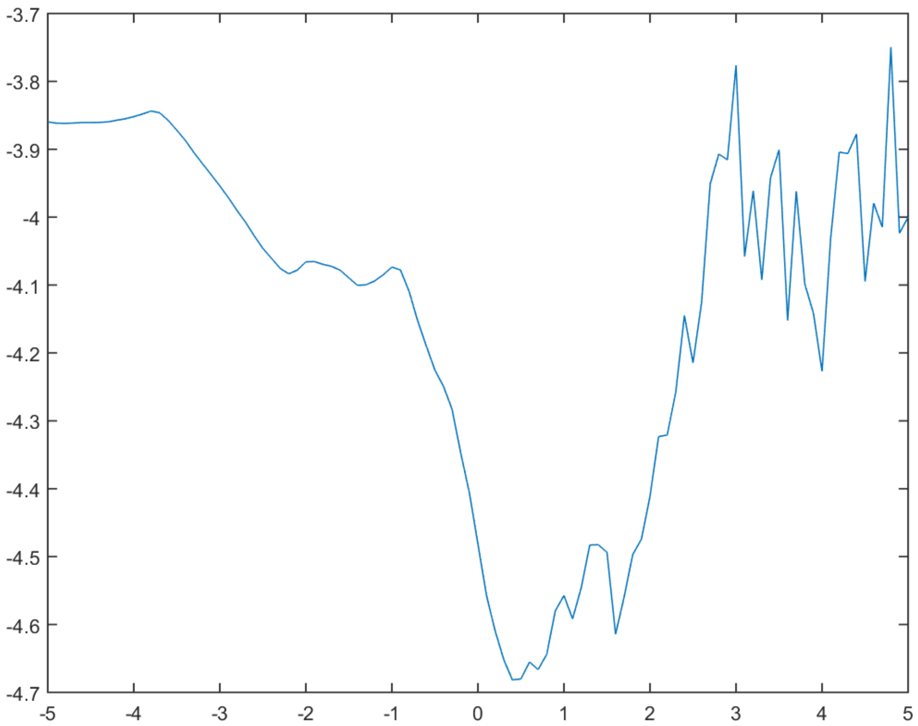


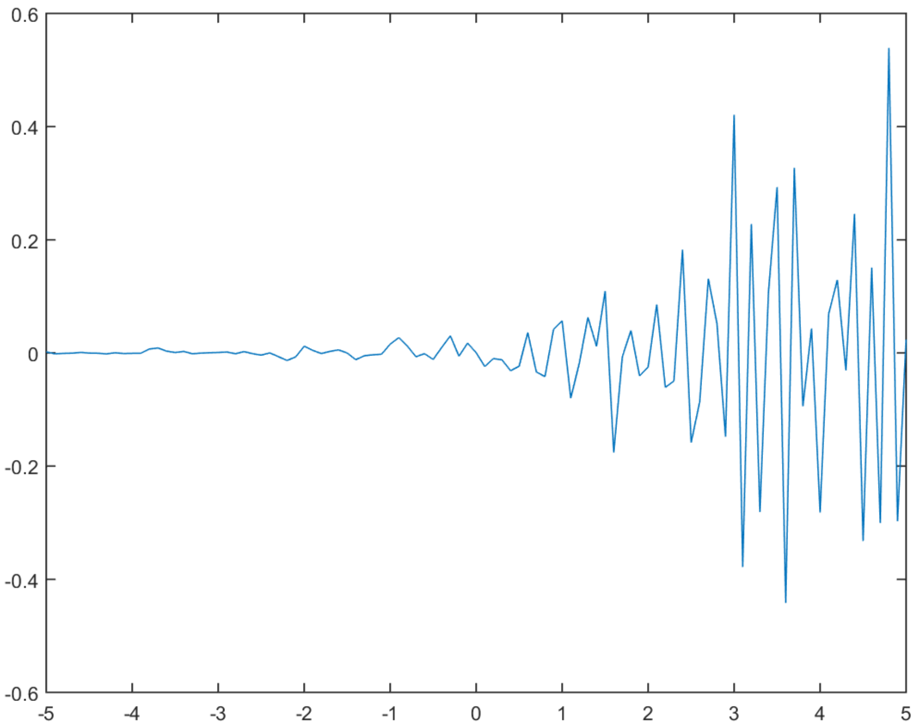


1D Convolution Examples

- Laplacian convolution approximates derivative:

$$w = [-1 \quad 2 \quad -1]$$

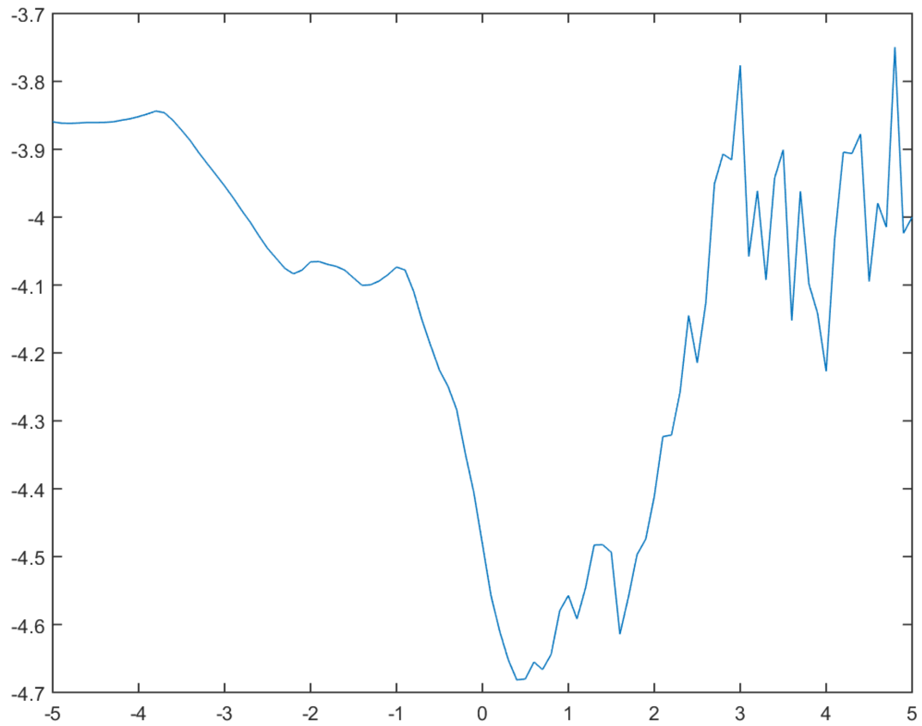


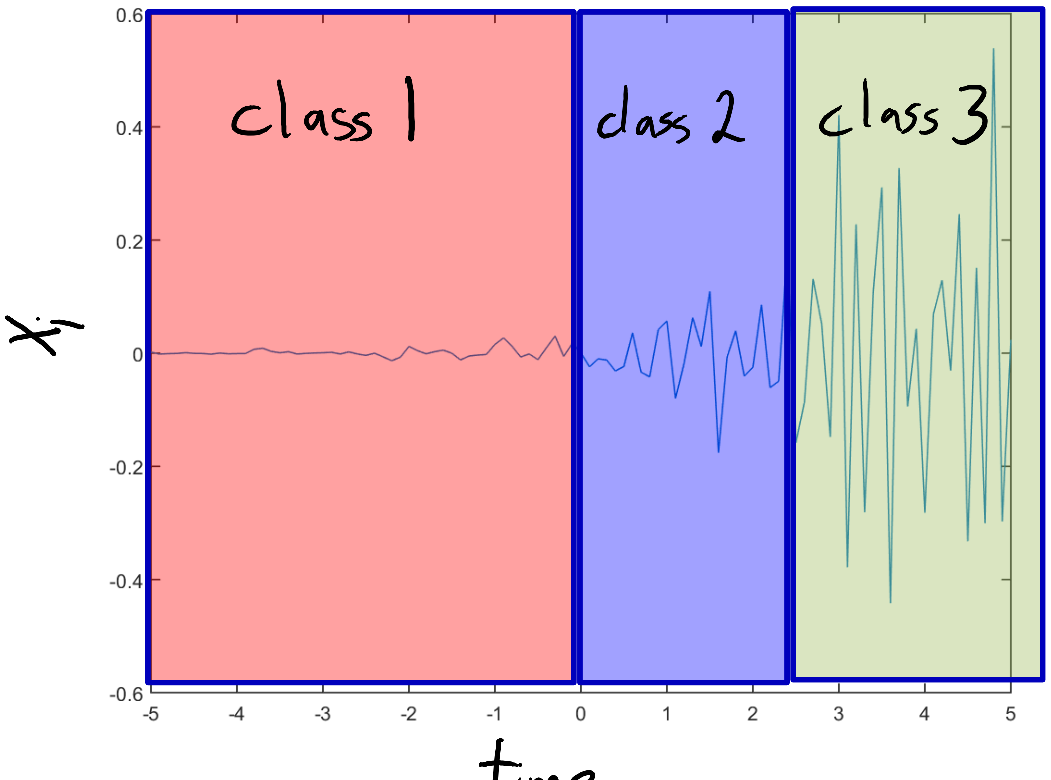


1D Convolution Examples

- Laplacian convolution approximates derivative:

$$w = [-1 \quad 2 \quad -1]$$





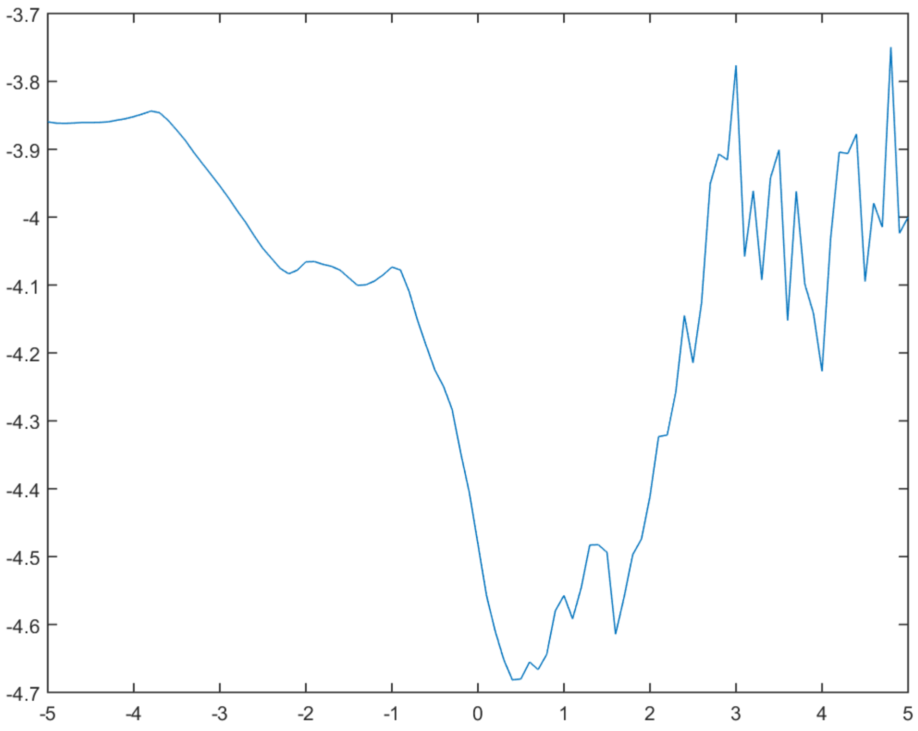
1D Convolution Examples

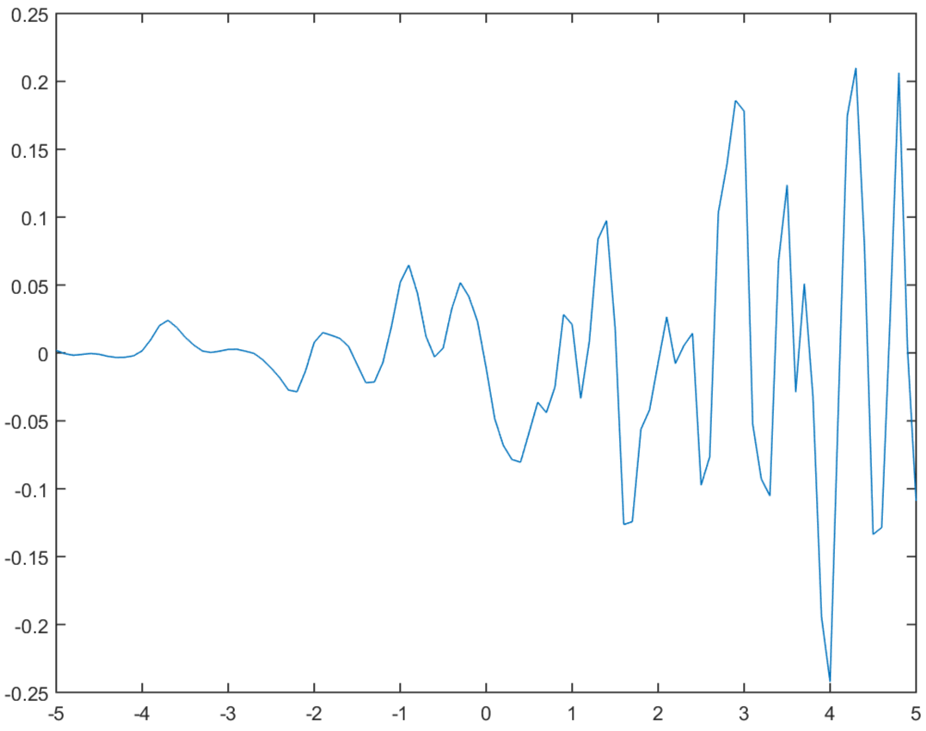
- Laplacian of Gaussian approximates derivative after blurring:

$$w_i = \left(1 - \frac{i^2}{2\sigma^2}\right) \exp\left(-\frac{i^2}{2\sigma^2}\right)$$

$w = [-0.1416\ \ -0.1781\ \ -0.2746\ \ 0.1640\ \ 0.8607\ \ 0.1640\ \ -0.2746\ \ -0.1781\ \ -0.1416]$

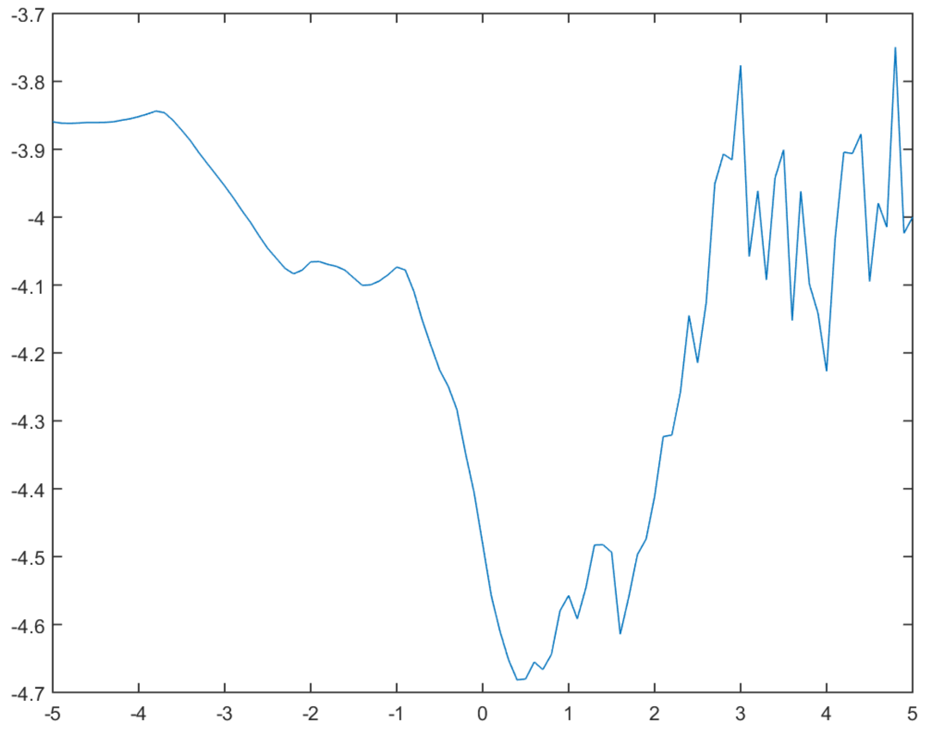
$(\sigma^2 = 1,\ m = 4)$

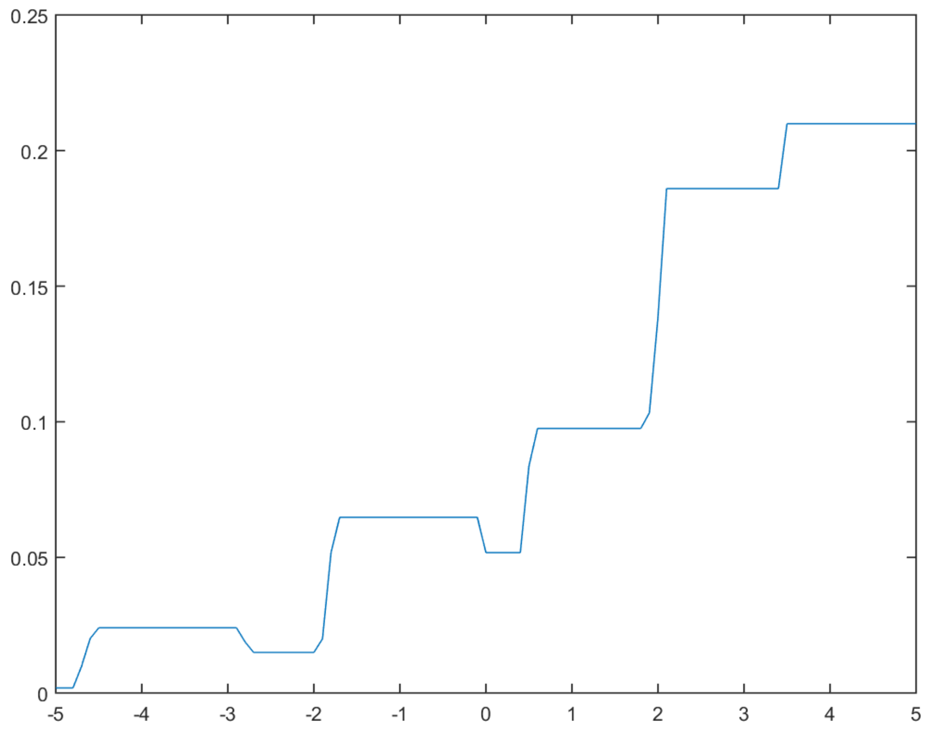




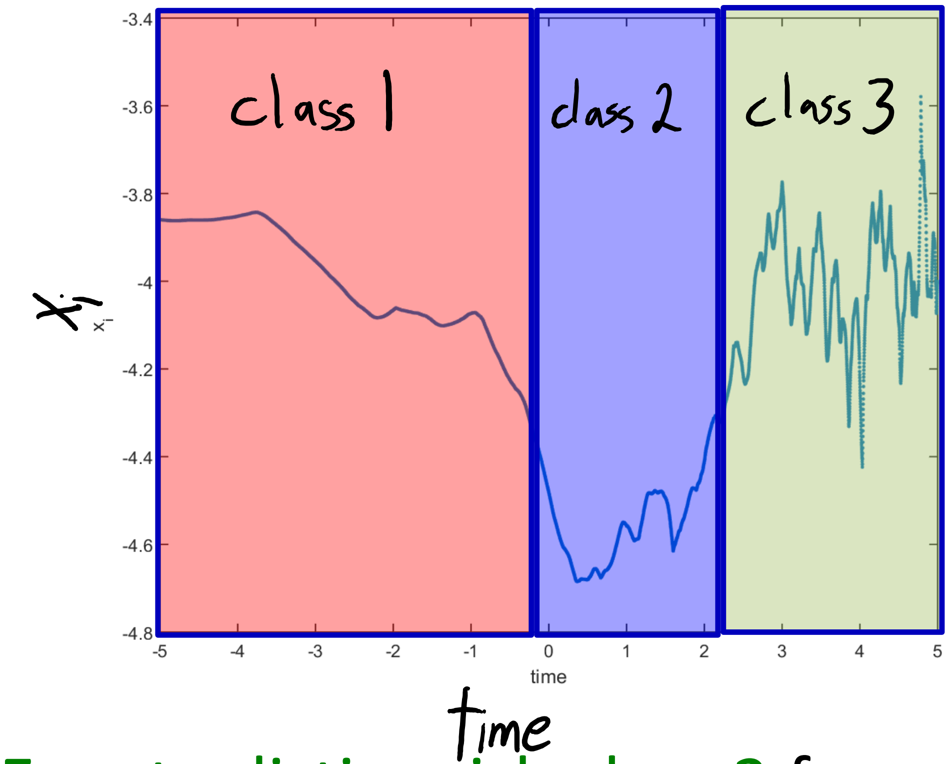
1D Convolution Examples

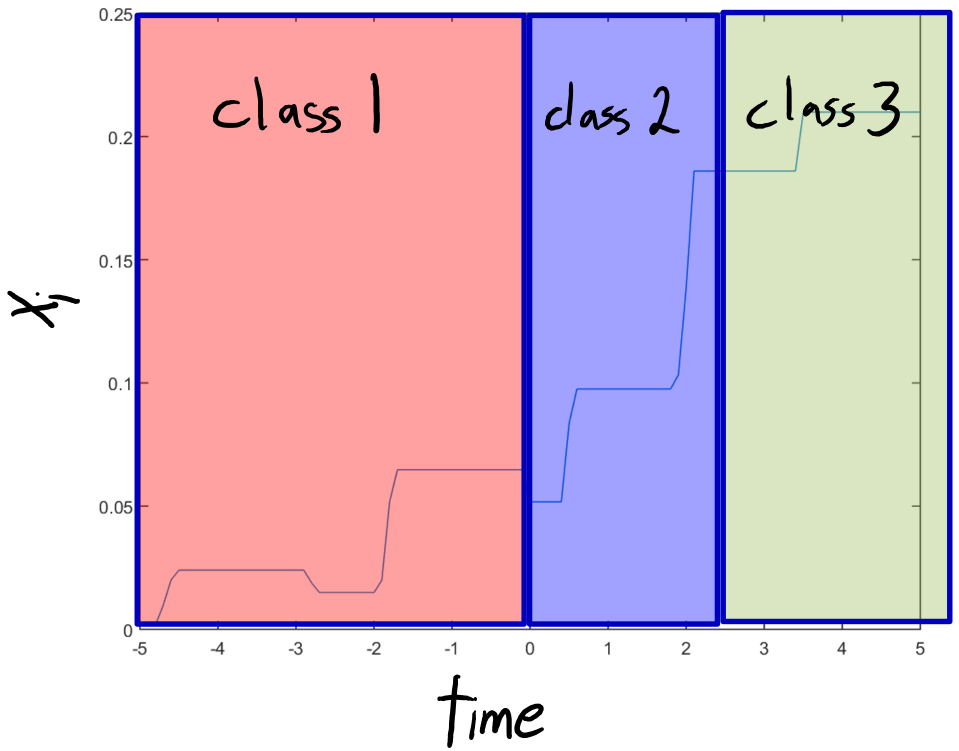
- We often use maximum over several convolutions as features:
  - We could take maximum of Laplacian of Gaussian over x_i and its 16 KNNs.





Why is this useful?





- Easy to distinguish class 2 from with original signal.
- Easy to distinguish class 1 from 3 with max(Laplacian(Gaussian)).
  - Convolutions and max(convolutions) are very useful for sequence data.
    - For sound data two related techniques are Fourier transforms and spectrograms.

Images and Higher-Order Convolution

- 2D convolution:
  - Signal 'x' is the pixel intensities in an 'n' by 'n' image.
  - Filter 'w' is the pixel intensities in a '2m+1' by '2m+1' image.
- The 2D convolution is given by:

$$z[i_1, i_2] = \sum_{j_1=-m}^{m} \sum_{j_2=-m}^{m} w[j_1, j_2]\, x[i_1 + j_1, i_2 + j_2]$$

- 3D and higher-order convolutions are defined similarly.

$$z[i_1, i_2, i_3] = \sum_{j_1=-m}^{m} \sum_{j_2=-m}^{m} \sum_{j_3=-m}^{m} w[j_1, j_2, j_3]\, x[i_1 + j_1, i_2 + j_2, i_3 + j_3]$$

Image Convolution Examples

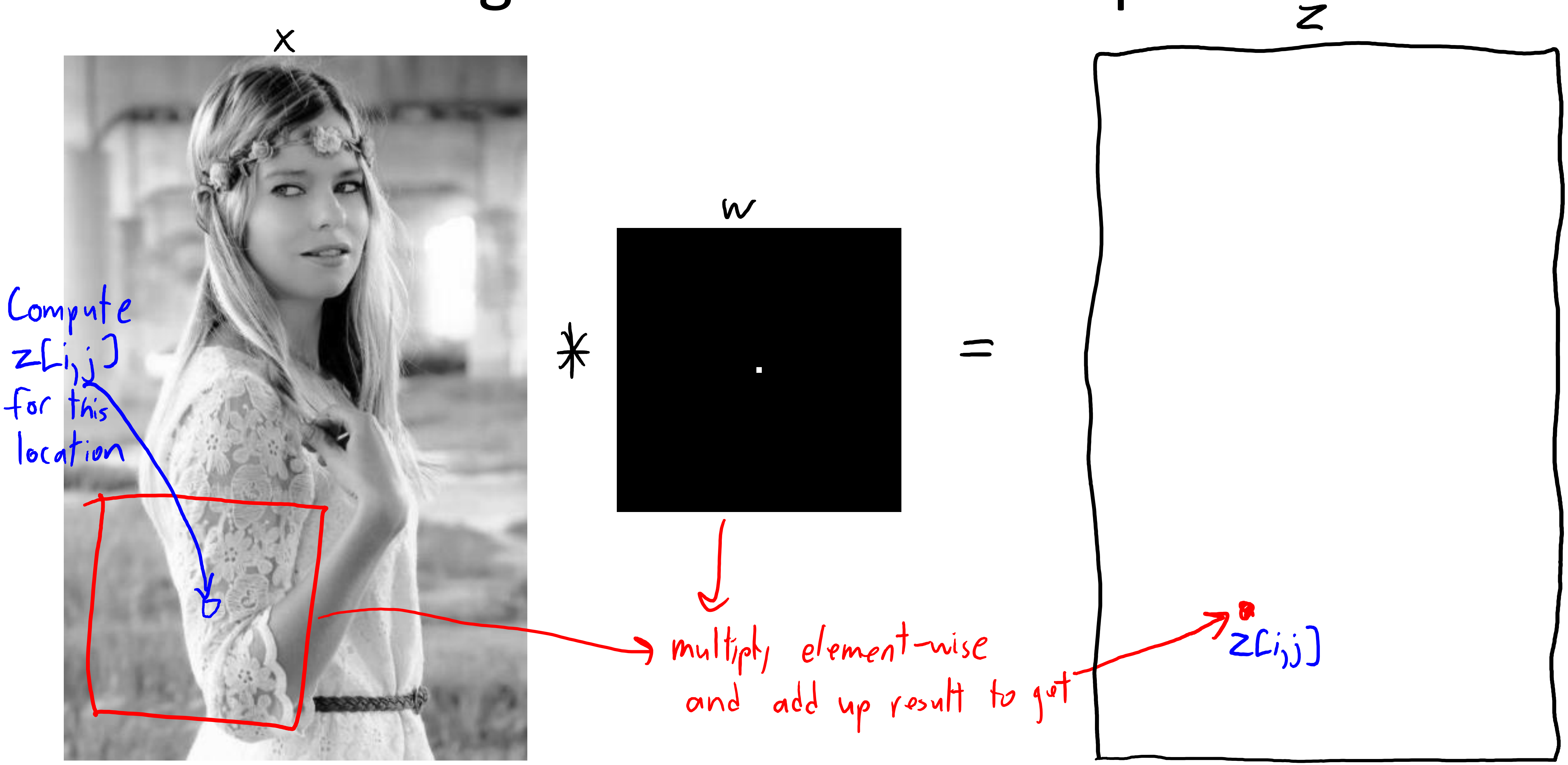
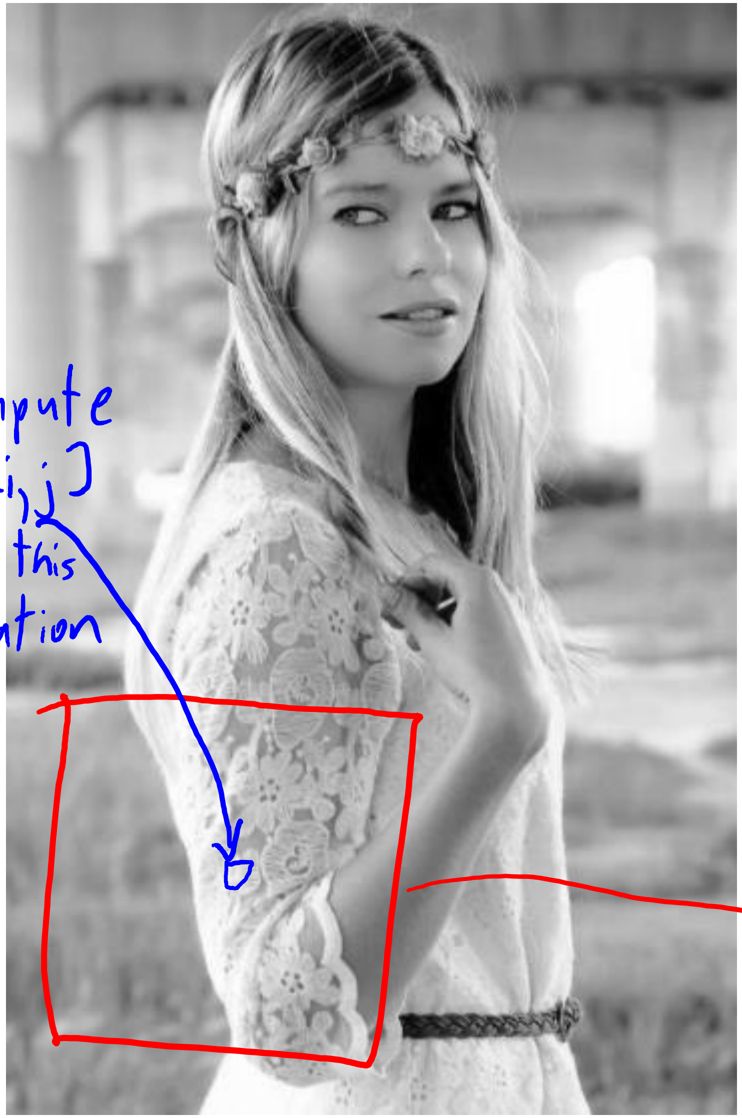


Image Convolution Examples

x

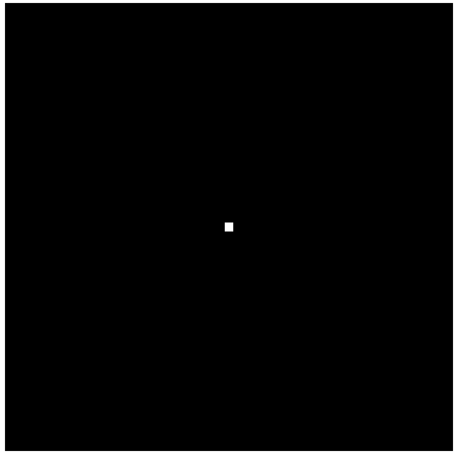
Compute z[i,j] for this location



Identity convolution: (zeroes with a '1' at $w_{0,0}$)

w

$*$



=

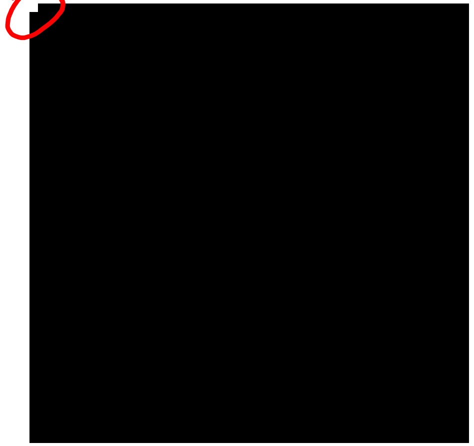
z

multiply element-wise and add up result to get

Image Convolution Examples

Translation Convolution:

$*$



Boundary: "zero"

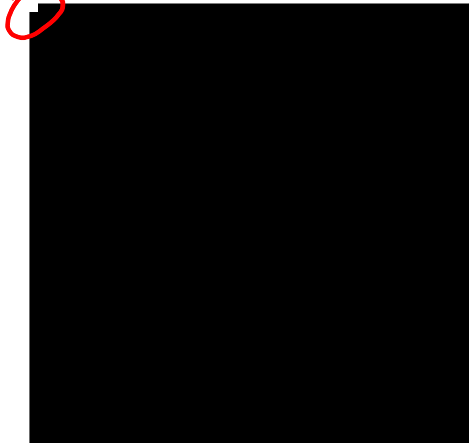
$=$

0

Image Convolution Examples

Translation Convolution:

$*$



Boundary: "replicate"

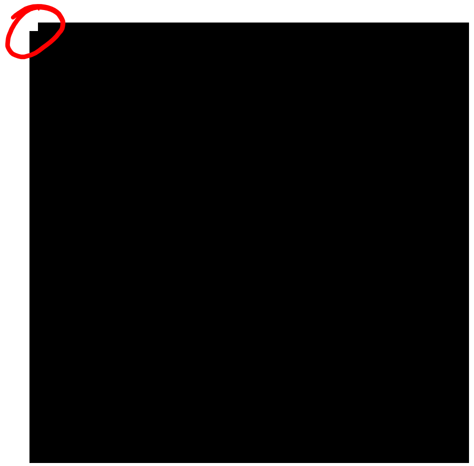
$=$

repeats

repeats

Image Convolution Examples

Translation Convolution:



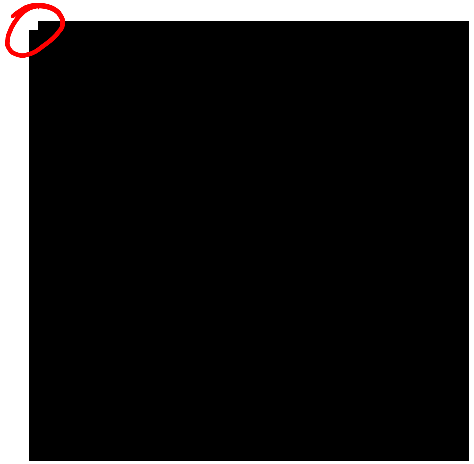
Boundary: "mirror"

flips

Image Convolution Examples

Translation Convolution:

$*$



$=$

Boundary: "ignore"

Image Convolution Examples

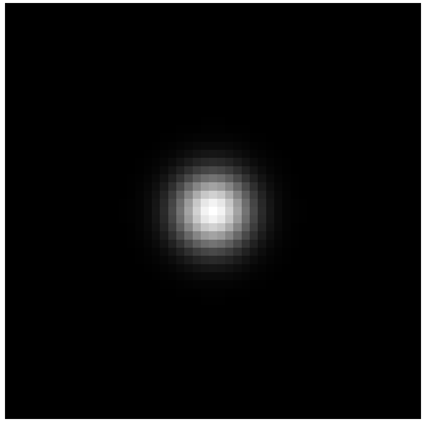
Average convolution:

$$* \frac{1}{51}\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 1 & 1 & \ddots & 1 \\ 1 & 1 & 1 & & 1 \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & 1 & 1 & \cdots & 1 \end{bmatrix} =$$

Image Convolution Examples

Gaussian Convolution:

$*$



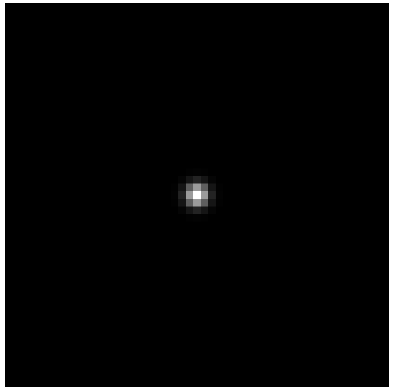
$=$

(smooths image)

Image Convolution Examples

Gaussian Convolution:

$*$



(smaller variance)

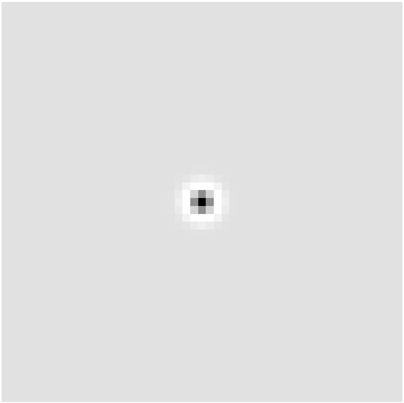
$=$

(<u>smooths</u> image)

Image Convolution Examples

Laplacian of Gaussian

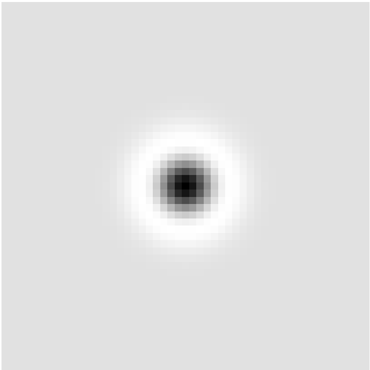
$*$



$=$

Image Convolution Examples

Laplacian of Gaussian

$*$  $=$

(larger variance)

Similar preprocessing may be done in basal ganglia and LGN.

Image Convolution Examples

"Emboss" filter:

$$* \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix} =$$

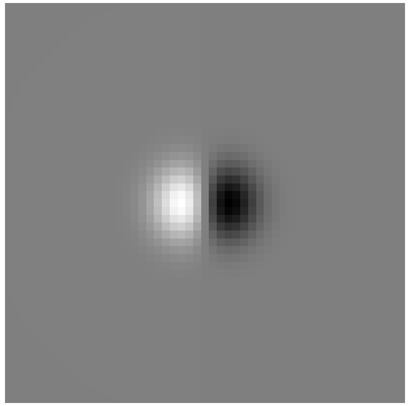
Many Photoshop effects are just convolutions.

http://setosa.io/ev/image-kernels

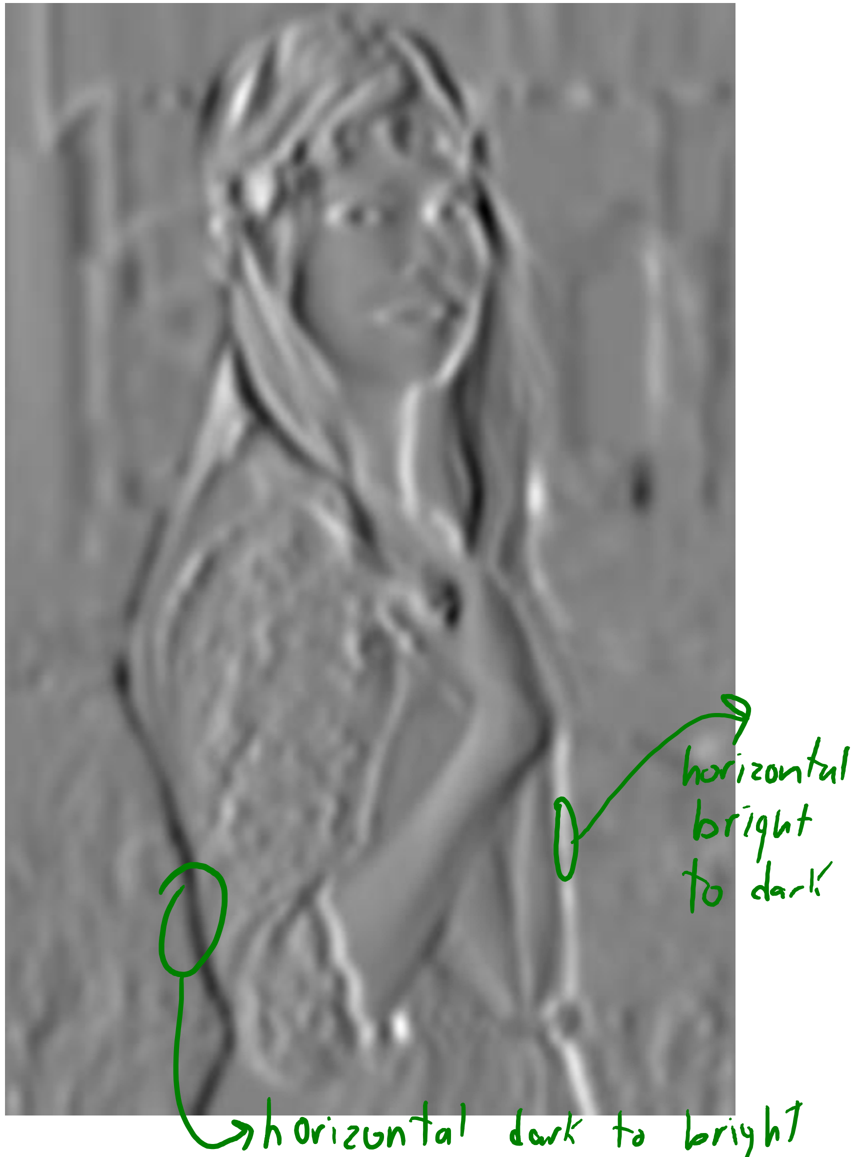
Image Convolution Examples

Gabor Filter
(Gaussian multiplied by sine or cosine)

$*$



$=$



$\parallel$

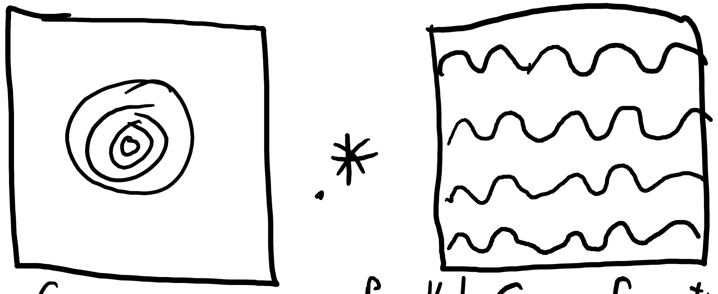
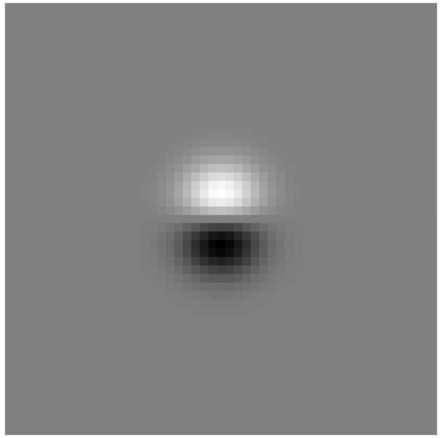


Image Convolution Examples

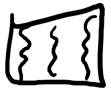
Gabor Filter
(Gaussian multiplied by sine or cosine)

$*$



$=$

Different orientations of the sine/cosine let us detect changes with different

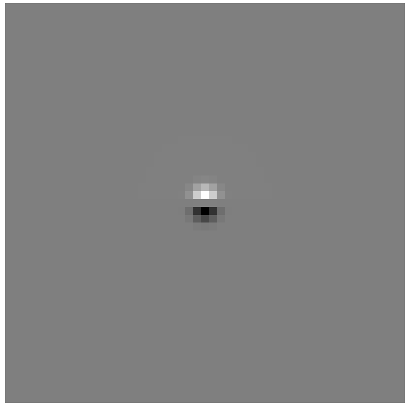


orientations.

Image Convolution Examples

Gabor Filter
(Gaussian multiplied by sine or cosine)

$*$



(smaller variance)

$=$

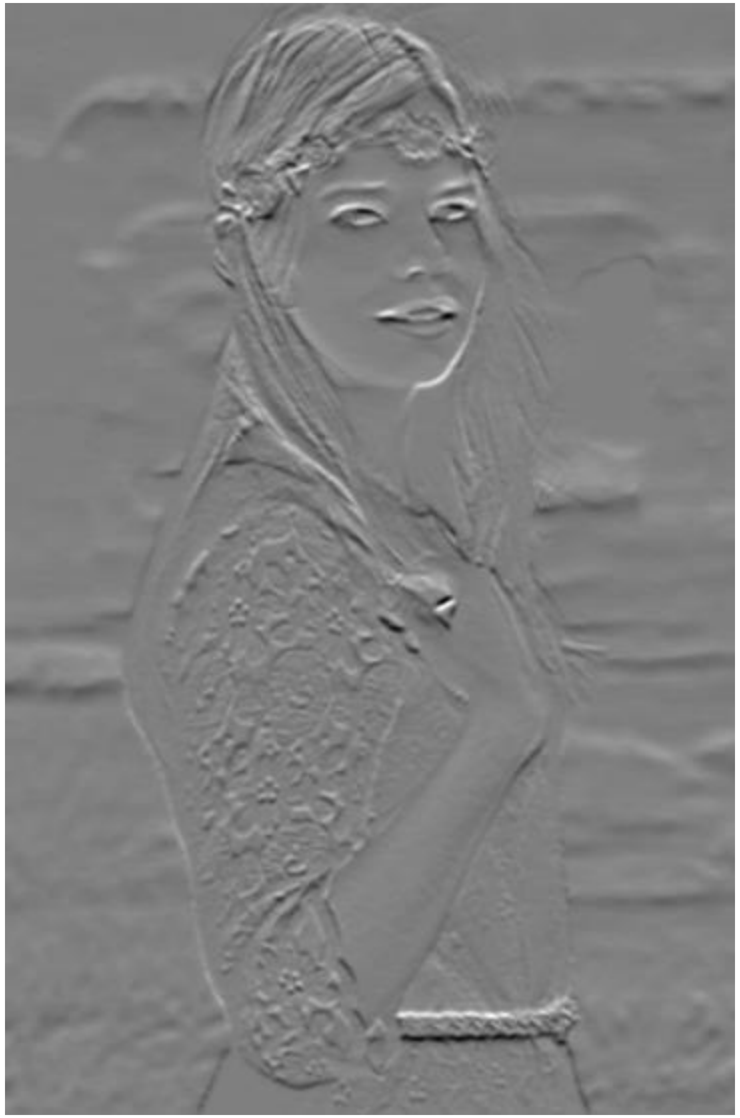
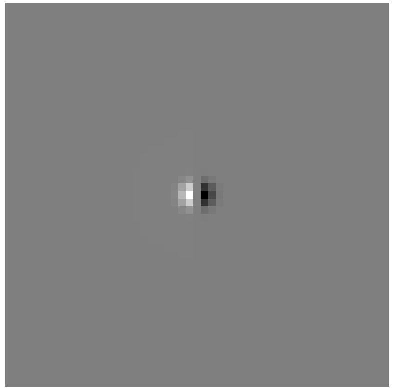


Image Convolution Examples

Gabor Filter
(Gaussian multiplied by sine or cosine)

$*$



$=$

(smaller variance)

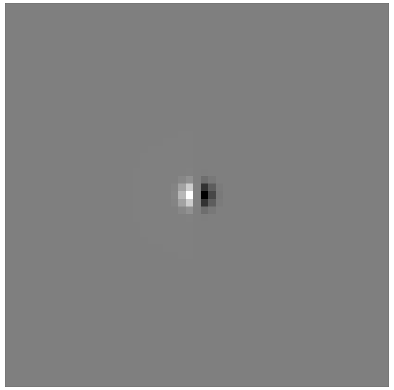
Vertical orientation

- Can obtain other orientations by rotating.
- May be similar to effect of V1 "simple cells."

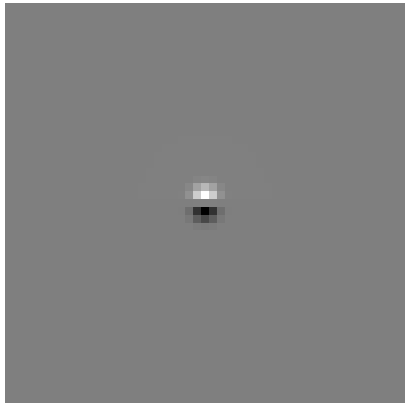
Image Convolution Examples

Max absolute value between horizontal and vertical Gabor:

*



*



maximum absolute value

"Horizontal/vertical edge detector"

3D Convolution

Represent as RGB

Can apply 3D convolutions

3D Convolution

Gaussian filter

3D Convolution

Gaussian filter

(higher variance on <u>green</u> channel)

3D Convolution

Sharpen the blue channel.

3D Convolution

Gabor filter on each channel.

Motivation for Convolutional Neural Networks

- Consider training neural networks on 256 by 256 images.
- Each z_i in first layer has 65536 parameters (and 3x this for colour).
  - We want to avoid this huge number (due to storage and overfitting).

- Key idea: make Wx_i act like convolutions (to make it smaller):
  - Each row of W only applies to part of x_i.
  - Use the same parameters between rows.

$$W_1 = [0 \quad 0 \quad 0 \; — w — \; 0 \; 0 \; 0]$$

$$W_2 = [0 \; — w — \; 0 \; 0 \; 0 \; 0 \; 0]$$

Convolutional Neural Networks

- Convolutional Neural Networks classically have 3 layer "types":
  - Fully connected layer: usual neural network layer with unrestricted W.

$$W^{(m)} = \begin{bmatrix} \text{---} \; w_1^{(m)} \; \text{---} \\ \text{---} \; w_2^{(m)} \; \text{---} \\ \vdots \\ \text{---} \; w_k^{(m)} \; \text{---} \end{bmatrix}$$

Convolutional Neural Networks

- Convolutional Neural Networks classically have 3 layer "types":
  - Fully connected layer: usual neural network layer with unrestricted W.
  - Convolutional layer: restrict W to results of several convolutions.

1D example

distance between centers of convolution is called "stride"

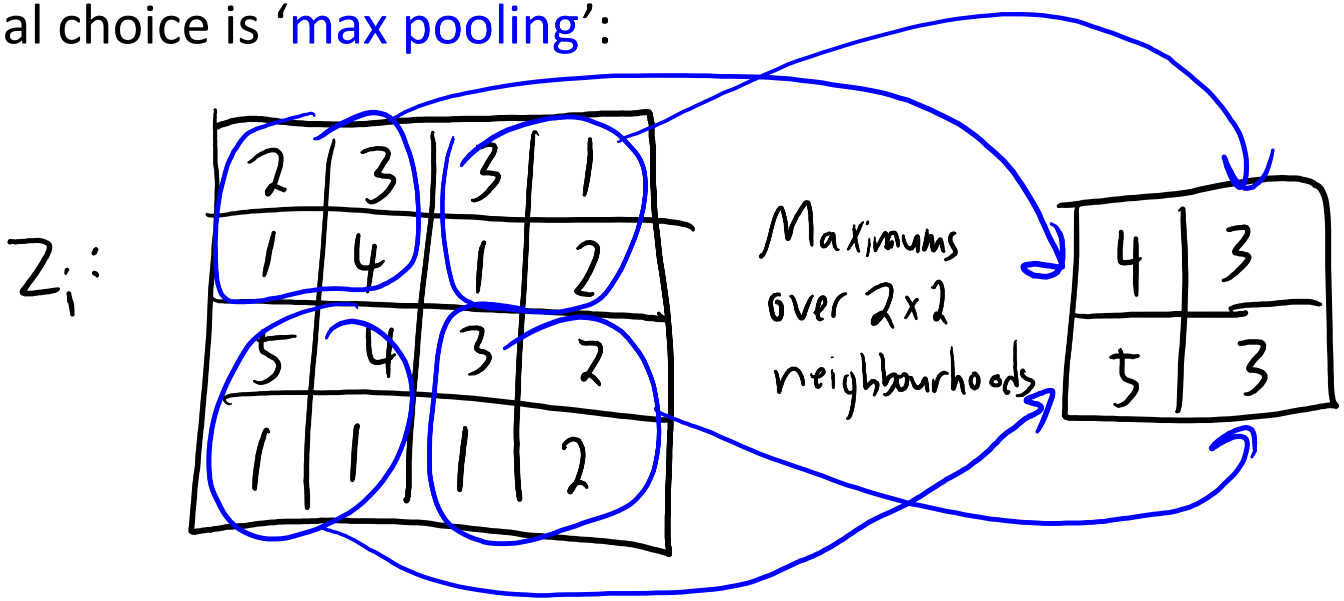
$$W^{(m)} = \begin{bmatrix} \text{——} & W_1^{(m)} & \text{——} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \text{——} & W_1^{(m)} & \text{——} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \text{——} & W_1^{(m)} & \text{——} \\ \text{——} & W_2^{(m)} & \text{——} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \text{——} & W_2^{(m)} & \text{——} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \text{——} & W_2^{(m)} & \text{——} \end{bmatrix}$$

Same $W_1^{(m)}$ used across multiple rows.

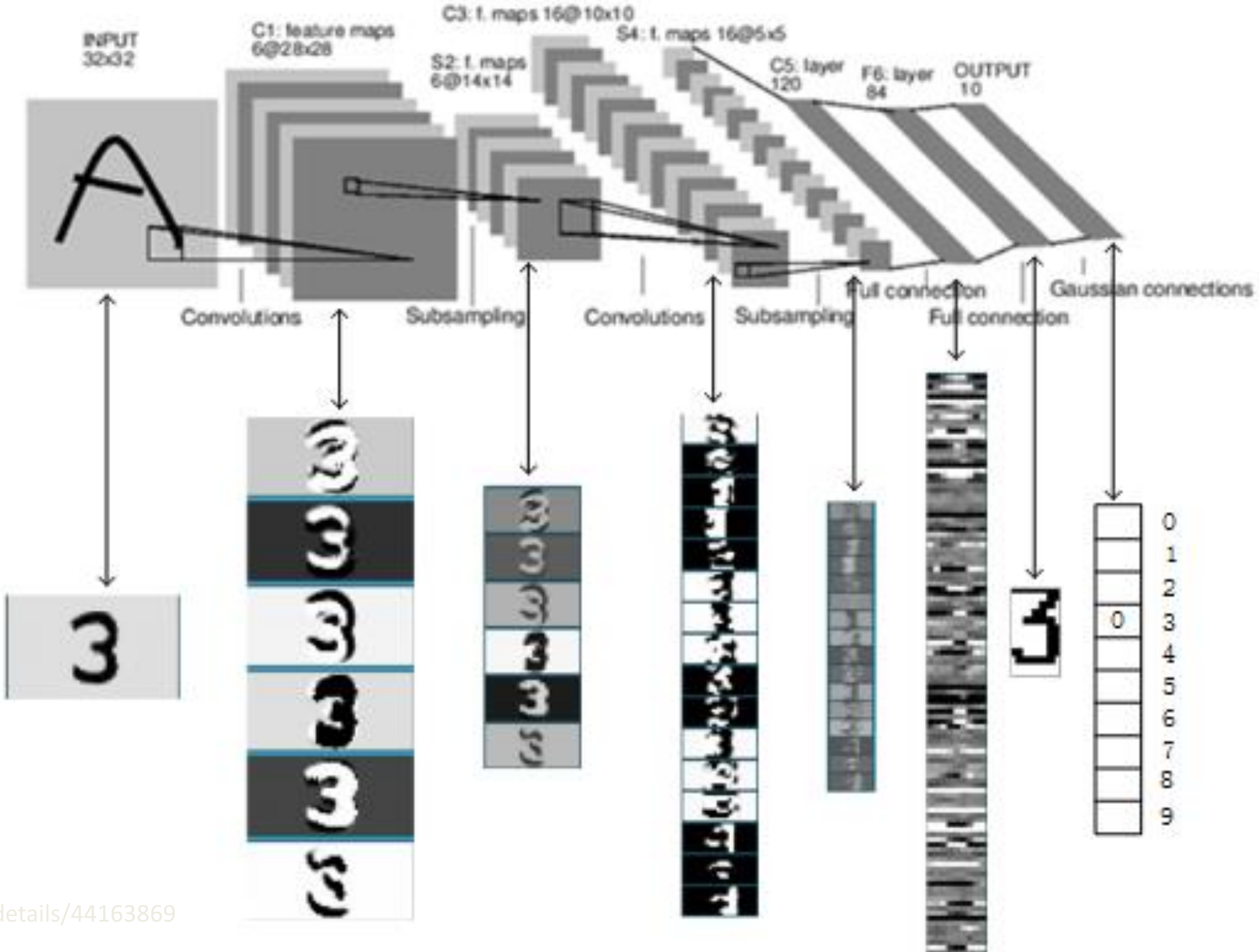
Sparse and small number of parameters

Convolutional Neural Networks

- Convolutional Neural Networks classically have 3 layer "types":
  - Fully connected layer: usual neural network layer with unrestricted W.
  - Convolutional layer: restrict W to results of several convolutions.
  - Pooling layer: downsamples result of convolution.
    - Can add invariances or just make the number of parameters smaller.
    - Usual choice is 'max pooling':



LeNet for Optical Character Recognition



Summary

- Convolutions are flexible class of signal/image transformations.
- Max(convolutions) can yield features that make classification easy.
- Convolutional neural networks:
  - Restrict W(m) matrices to represent sets of convolutions.
  - Often combined with max (pooling).

- Next time: modern convolutional neural networks and applications.
  - Image segmentation, depth estimation, image colorization, artistic style.