Deep Learning

Advanced Machine Learning
26 September 2024
Profs. Luigi Cinque, Fabio Galasso and Marco Raoul Marini

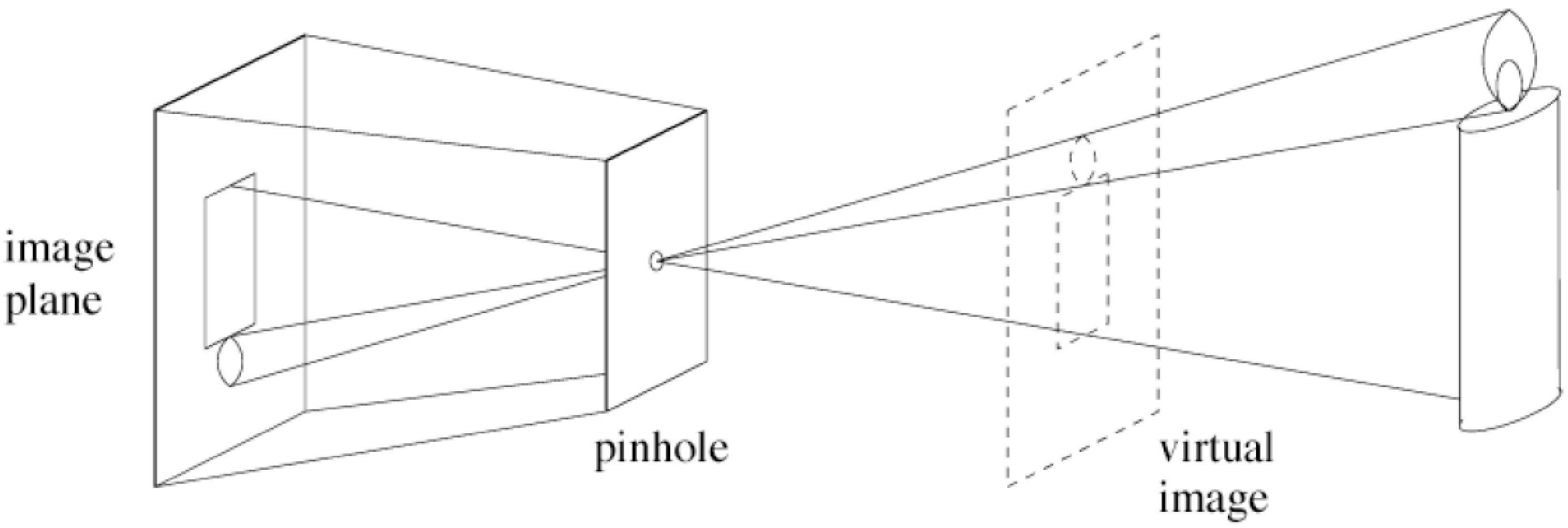
SAPIENZA
UNIVERSITÀ DI ROMA

Basics of Digital Imaging

Sapienza Università di Roma

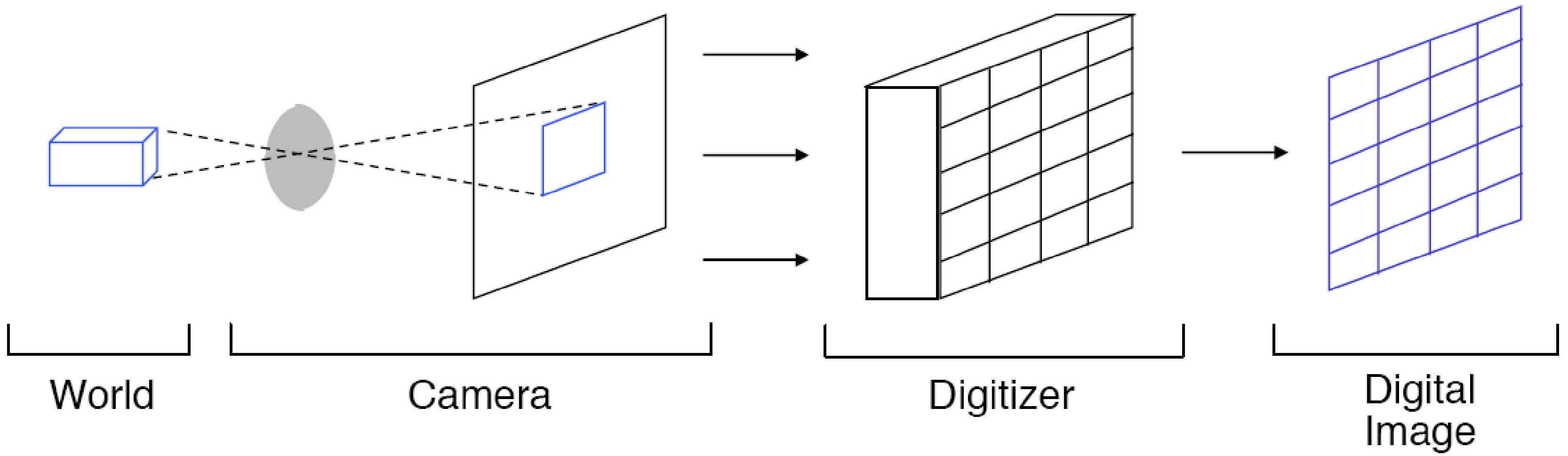
Pinhole Camera (Model)

- (simple) standard and abstract model today
  - box with a small hole in it



Digital Images

- Imaging Process:
  - (pinhole) camera model
  - digitizer to obtain digital image



Digital Image Processing

- Image Filtering
  - take some local image patch (e.g. 3x3 block)
  - image filtering: apply some function to local image patch

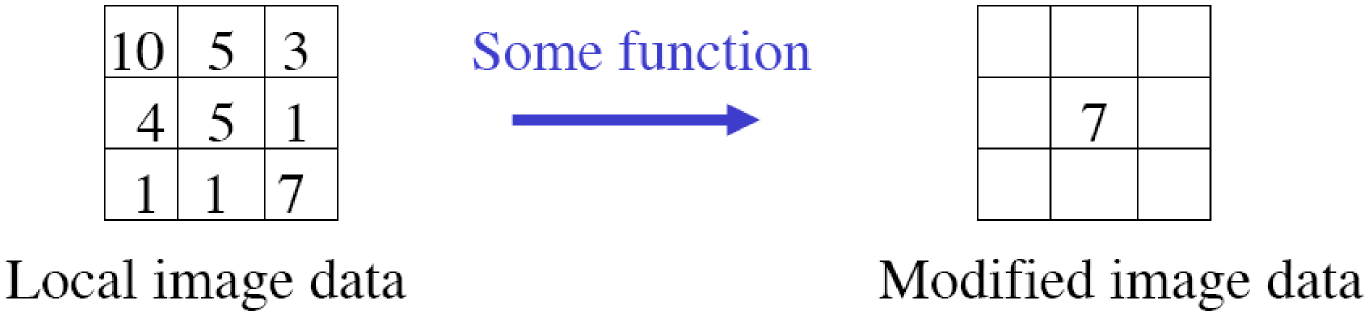


Image Filtering

- Some Examples:
  - what assumptions are you making to infer the center value?

3	3	3
3	?	3
3	3	3

3

3	4	3
2	?	5
5	4	2

3 or 4

Image Filtering: 2D Signals and Convolution

- Image Filtering
  - to reduce noise,
  - to fill-in missing values/information
  - to extract image features (e.g. edges/corners), etc.

$$\begin{array}{|c|c|c|}\hline 2 & 3 & 3 \\ \hline 3 & 20 & 2 \\ \hline 3 & 2 & 3 \\ \hline \end{array} \longrightarrow \begin{array}{|c|c|c|}\hline 2 & 3 & 3 \\ \hline 3 & 3 & 2 \\ \hline 3 & 2 & 3 \\ \hline \end{array}$$

- Simplest case:
  - linear filtering: replace each pixel by a linear combination of its neighbors

- 2D convolution (discrete):

$$f[m,n] = I \otimes g = \sum_{k,l} I[m-k, n-l] g[k,l]$$

  - discrete Image: *I[m,n]*
  - filter 'kernel': *g[k,l]*
  - 'filtered' image: *f[m,n]*

$$\underset{f[m,n]}{\begin{array}{|c|c|c|}\hline & & \\ \hline & 18 & \\ \hline & & \\ \hline \end{array}} = \underset{I[k,l]}{\begin{array}{|c|c|c|}\hline 8 & 5 & 2 \\ \hline 7 & 5 & 3 \\ \hline 9 & 4 & 1 \\ \hline \end{array}} \otimes \underset{g[k,l]}{\begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline \end{array}}$$

can be expressed as matrix multiplication!

Luigi Cinque, Fabio Galasso and Marco Raoul Marini

SAPIENZA UNIVERSITÀ DI ROMA

Image Filtering: 2D Signals and Convolution

- 2D convolution (discrete): $f[m,n] = I \otimes g = \sum_{k,l} I[m-k, n-l] g[k,l]$
  - discrete Image: *I[m,n]*
  - filter 'kernel': *g[k,l]*
  - 'filtered' image: *f[m,n]*

$$f[m,n] \quad I[k,l] \quad g[k,l]$$

$$\begin{array}{|c|c|c|} \hline & & \\ \hline & 18 & \\ \hline & & \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline 8 & 5 & 2 \\ \hline 7 & 5 & 3 \\ \hline 9 & 4 & 1 \\ \hline \end{array} \otimes \begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline \end{array}$$

l k (m, n) l k

$$= \sum_{\substack{-1 < k < +1 \\ -1 < l < +1}} I[m-k, n-l] g[k,l]$$ (m, n)

- mirror the filter (k and l)
- swipe it across the image
- multiply and sum

$$= I[m+1, n+1] g[-1,-1] \qquad (k=-1, l=-1)$$

$$+ I[m+1, n] g[-1, 0] \qquad (k=-1, l=0)$$

$$+ I[m+1, n-1] g[-1, +1] \qquad (k=-1, l=+1)$$

$$+ \ldots$$

Luigi Cinque, Fabio Galasso and Marco Raoul Marini

Image Filtering: 2D Signals and Convolution

- 2D convolution (discrete): $f[m,n] = I \otimes g = \sum_{k,l} I[m-k, n-l]g[k,l]$
  - discrete Image: *I[m,n]*
  - filter ‘kernel’: *g[k,l]*
  - ‘filtered’ image: *f[m,n]*

$$\underset{f[m,n]}{\begin{array}{|c|c|c|}\hline & & \\ \hline & 18 & \\ \hline & & \\ \hline \end{array}} = \underset{I[k,l]}{\begin{array}{|c|c|c|}\hline 8 & 5 & 2 \\ \hline 7 & 5 & 3 \\ \hline 9 & 4 & 1 \\ \hline \end{array}} \otimes \underset{g[k,l]}{\begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline \end{array}}$$

- special case:
  - convolution (discrete) of a 2D-image with a 1D-filter

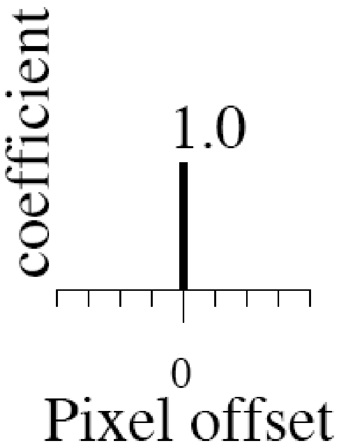
$$f[m,n] = I \otimes g = \sum_{k} I[m-k, n]g[k]$$

$$\underset{g[k]}{\begin{array}{|c|}\hline -1 \\ \hline 0 \\ \hline 1 \\ \hline \end{array}}$$

Linear Filtering (warm-up slide)

$$f[m,n] = I \otimes g = \sum_{k} I[m-k,n]g[k]$$

original



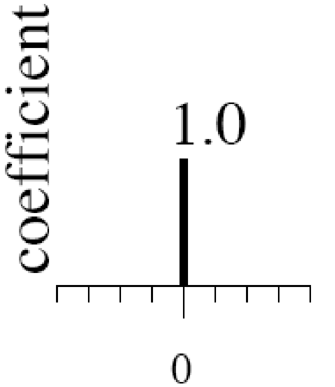
$= ?$

$I \quad \otimes \quad g \quad = f$

Linear Filtering (warm-up slide)

$$f[m,n] = I \otimes g = \sum_k I[m-k,n]g[k]$$

original

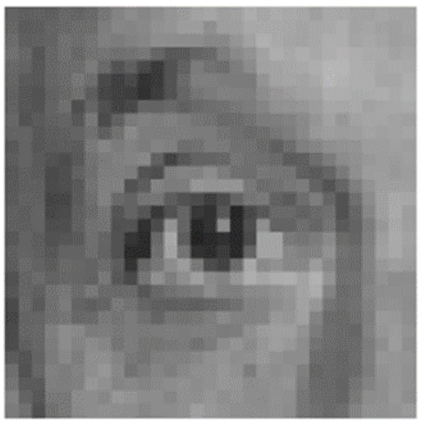


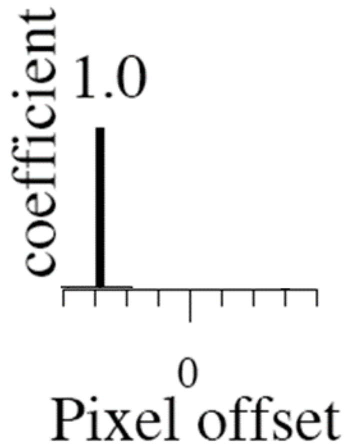
Filtered
(no change)

$$I \qquad \otimes \qquad g \qquad = f$$

Linear Filtering

$$f[m,n] = I \otimes g = \sum_k I[m-k,n]g[k]$$





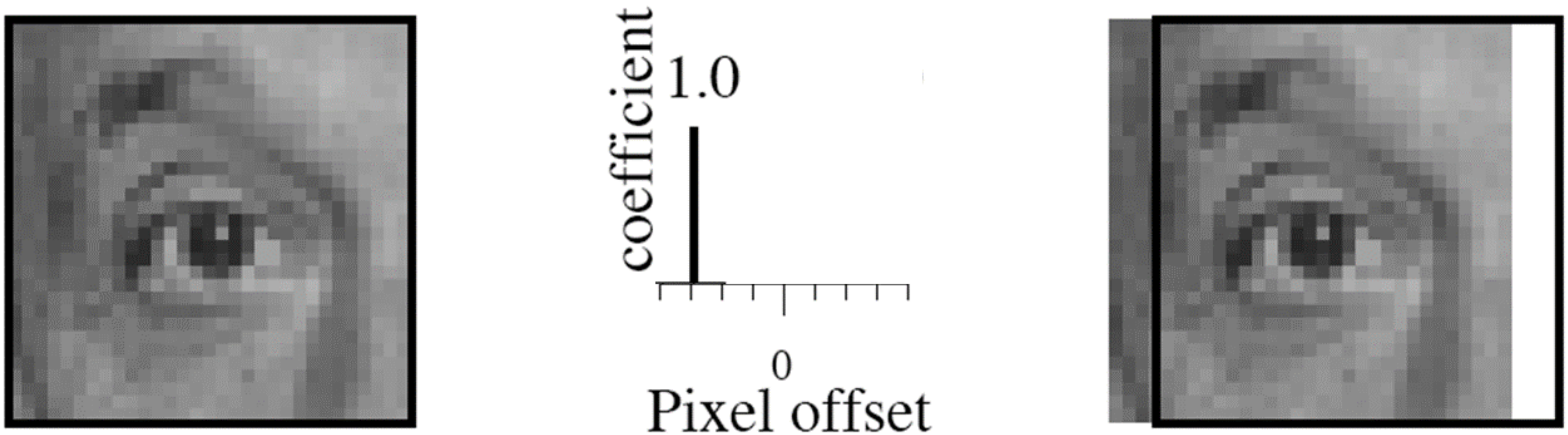
?

original

I $\otimes$ g $= f$

Linear Filtering

$$f[m,n] = I \otimes g = \sum_{k} I[m-k,n]g[k]$$



original

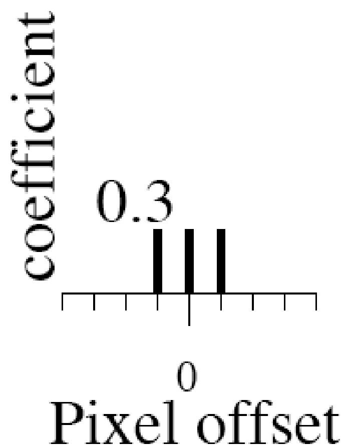
shifted

$I \quad \otimes \quad g \quad = f$

Linear Filtering

$$f[m,n] = I \otimes g = \sum_k I[m-k,n]g[k]$$

original



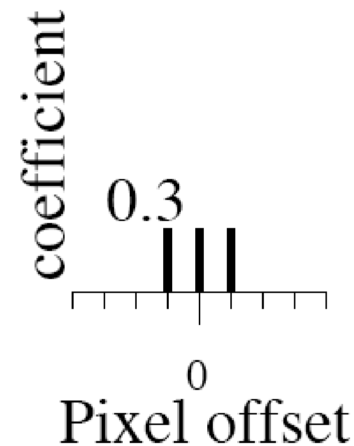
?

$I \quad \otimes \quad g \quad = f$

Blurring

$$f[m,n] = I \otimes g = \sum_k I[m-k,n]g[k]$$

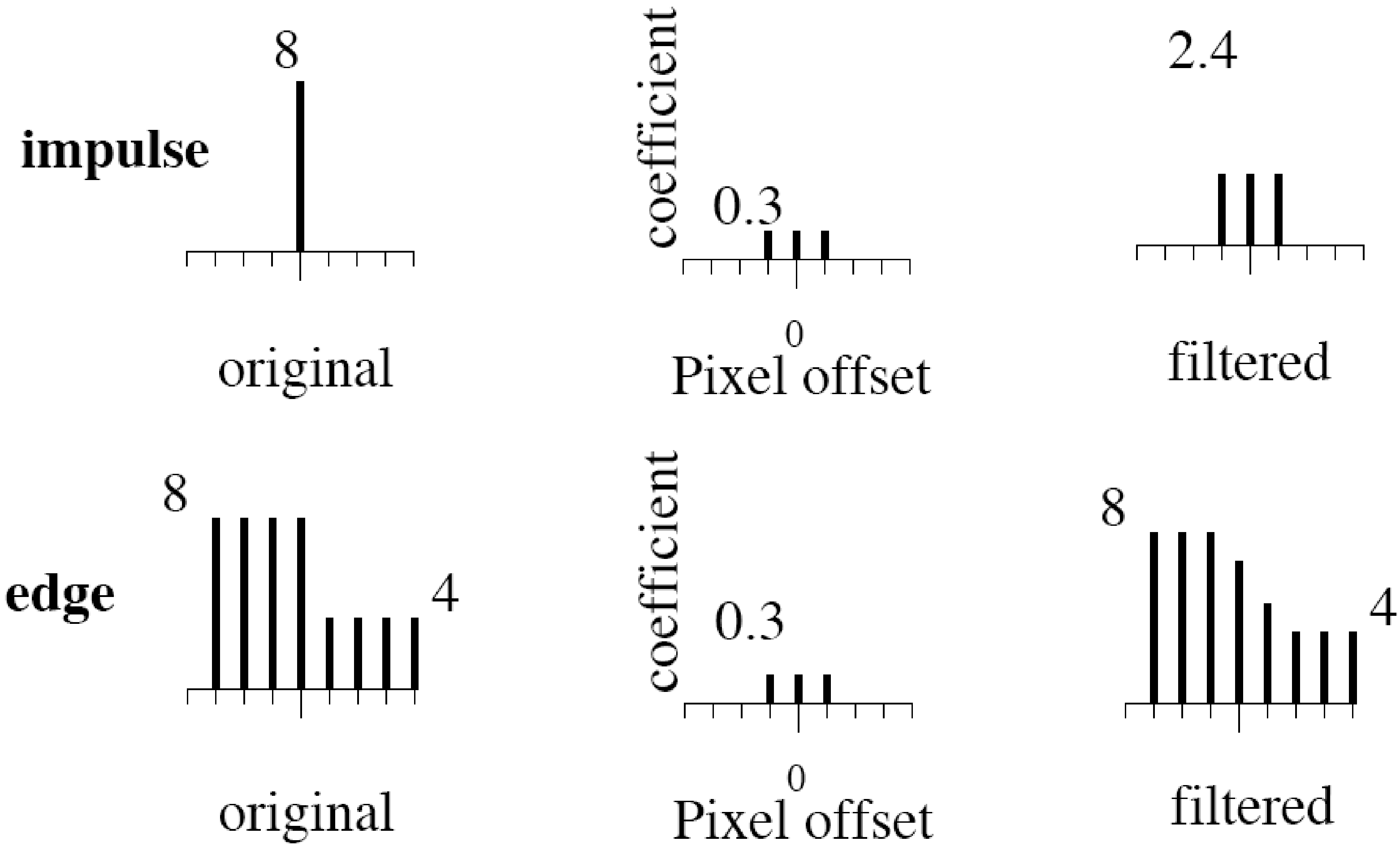
original



Blurred (filter applied in both dimensions).

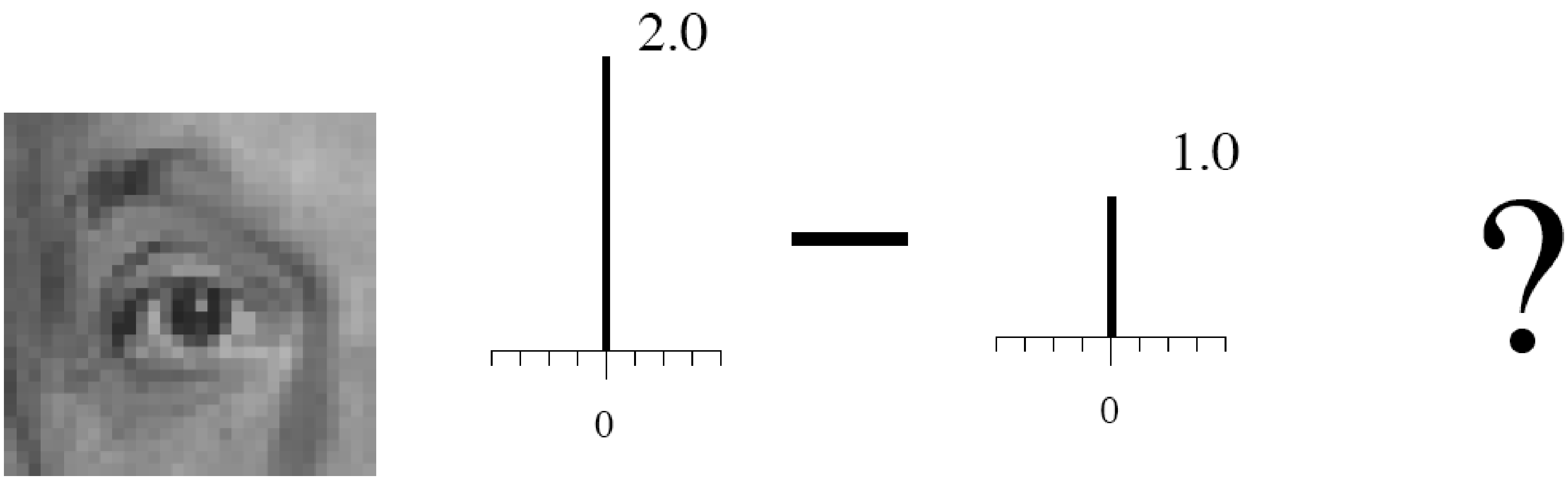
$I \qquad \otimes \qquad g$

Blurring Examples



Linear Filtering (warm-up slide)

$$f[m,n] = I \otimes g_1 - I \otimes g_2 = I \otimes (g_1 - g_2)$$

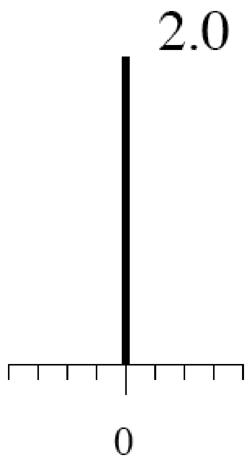


original

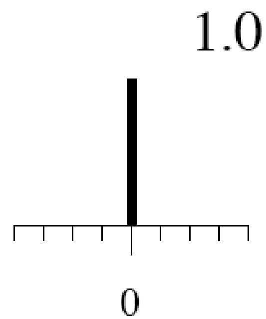
Linear Filtering (warm-up slide)

$$f[m,n] = I \otimes g_1 - I \otimes g_2 = I \otimes (g_1 - g_2)$$

original



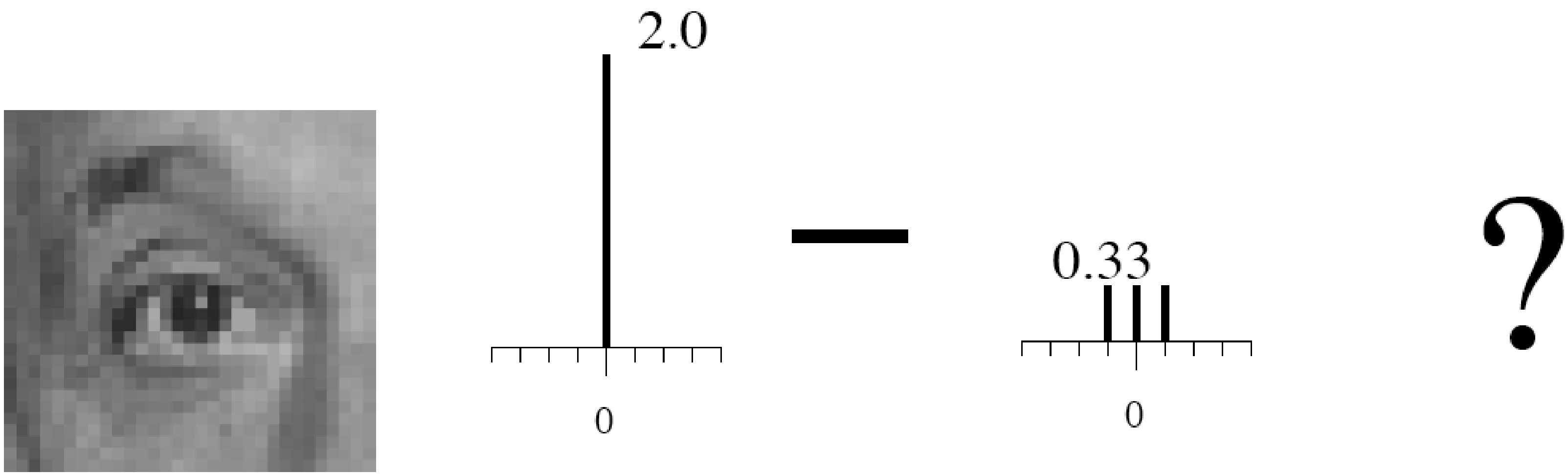
−



Filtered
(no change)

Linear Filtering

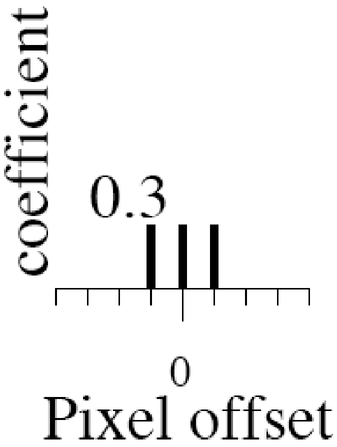
$$f[m,n] = I \otimes g_1 - I \otimes g_2 = I \otimes (g_1 - g_2)$$



original

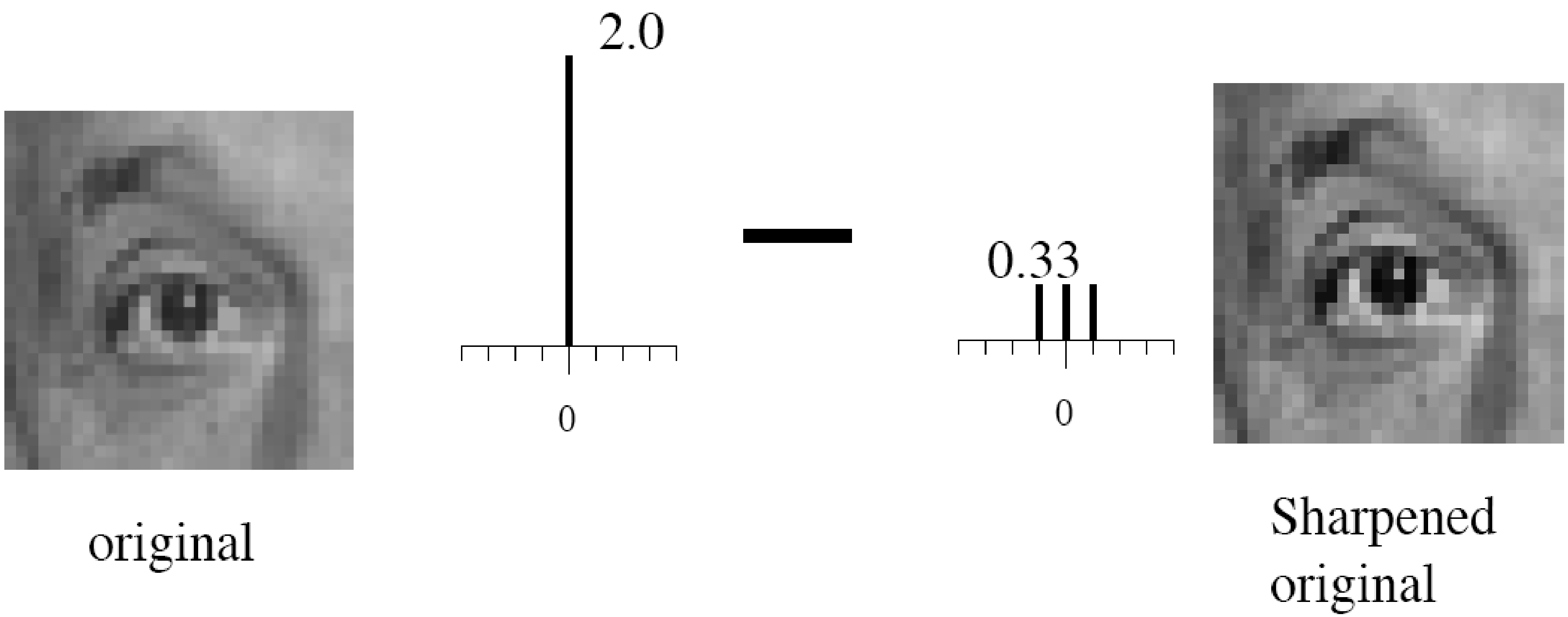
(remember blurring)

original

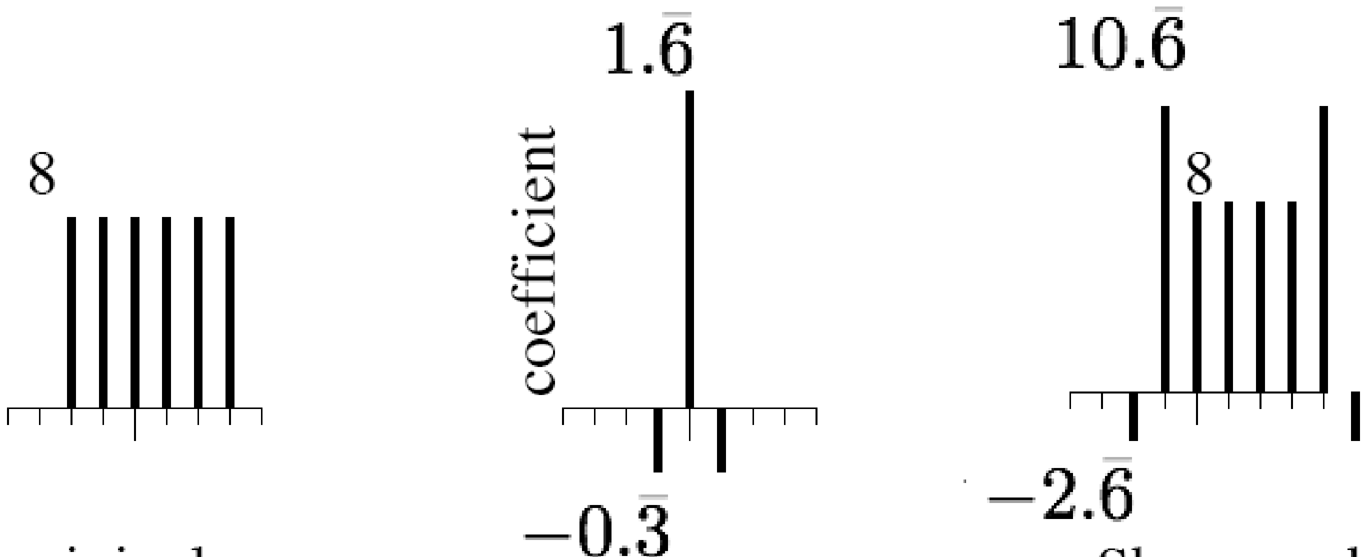


Blurred (filter applied in both dimensions).

Sharpening



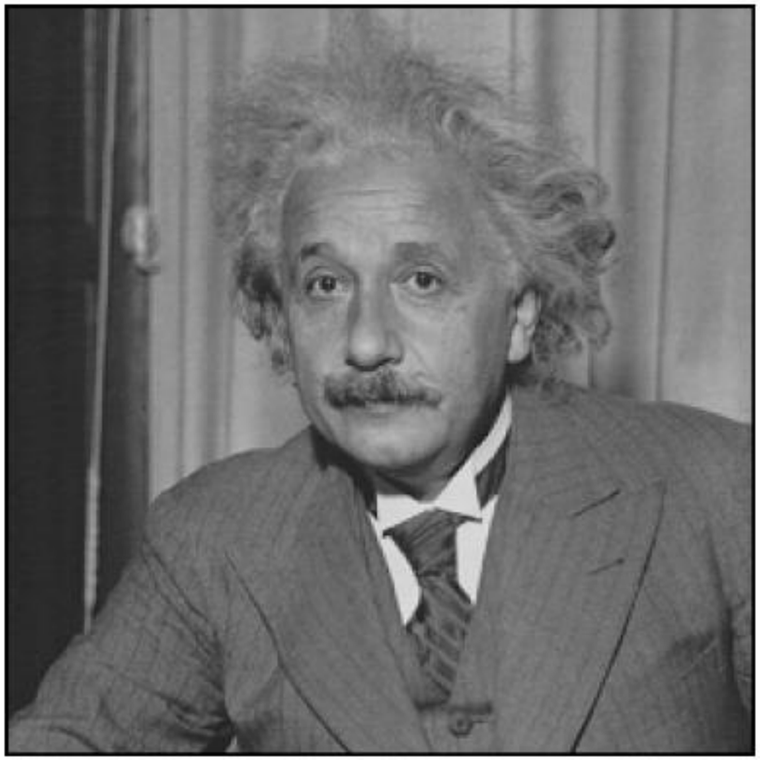
Sharpening Example



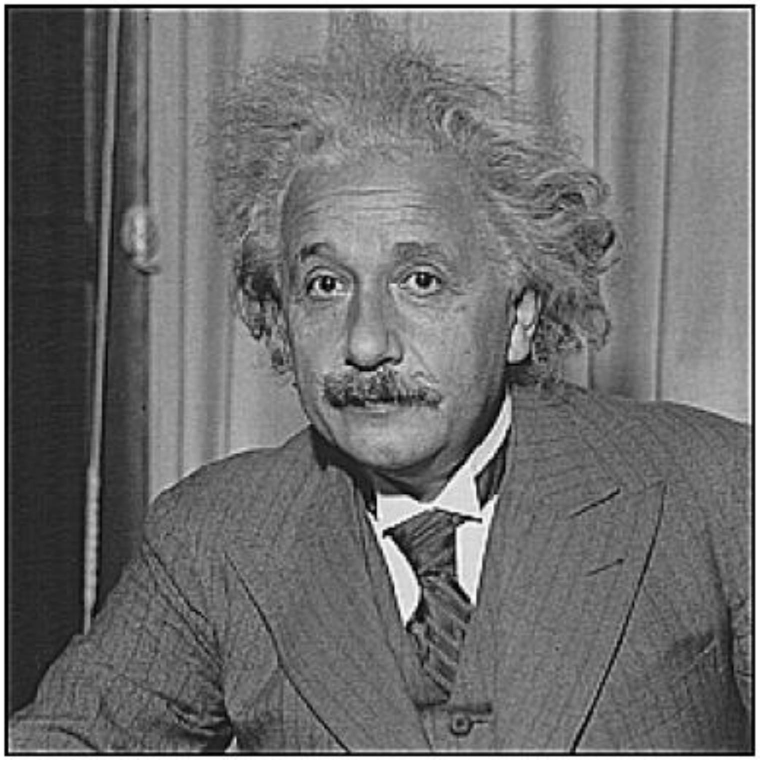
original

Sharpened
(differences are
accentuated; constant
areas are left untouched).

Sharpening



before



after

Linear Systems

- Basic Properties:
  - homogeneity $T[a X] = a T[X]$
  - additivity $T[X_1 + X_2] = T[X_1] + T[X_2]$
  - superposition $T[aX_1 + bX_2] = a T[X_1] + b T[X_2]$
  - linear systems <=> superposition

- examples:
  - matrix operations (additions, multiplication)
  - convolutions

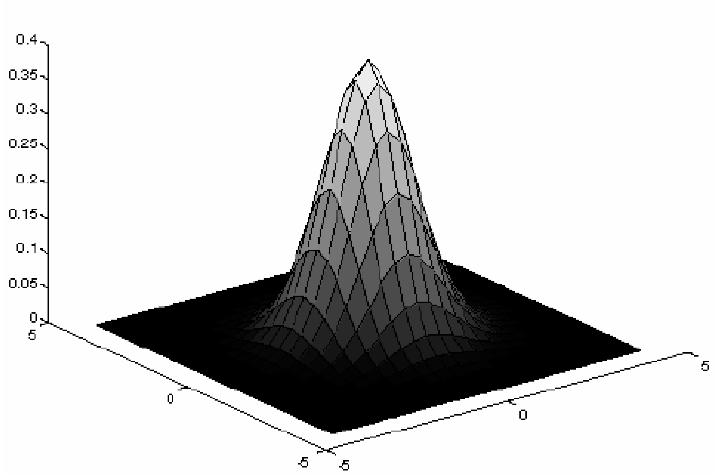
Average Filter

- Average Filter
  - replaces each pixel with an average of its neighborhood
  - Mask with positive entries that sum to 1
- if all weights are equal, it is called a BOX filter

$$\begin{array}{|c|c|c|}\hline \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \hline \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \hline \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \hline \end{array}$$

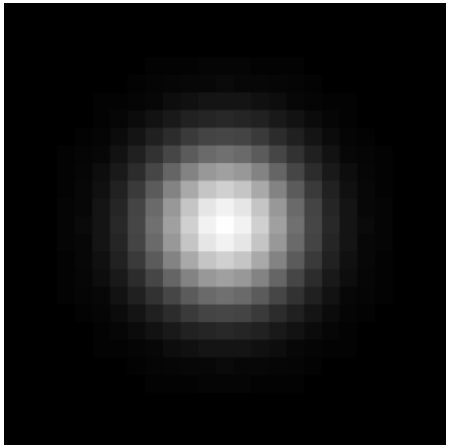
Gaussian Averaging (An Isotropic Gaussian)

- Rotationally symmetric
- Weights nearby pixels more than distant ones
  - this makes sense as 'probabilistic' inference



- the pictures show a smoothing kernel proportional to

$$g(x,y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$



Smoothing with a Gaussian

- Example:

Original Image Gaussian-filtered Box-filtered

Smoothing with a Gaussian

- Another Example:

Original image

Gaussian Blur applied

Efficient Implementation

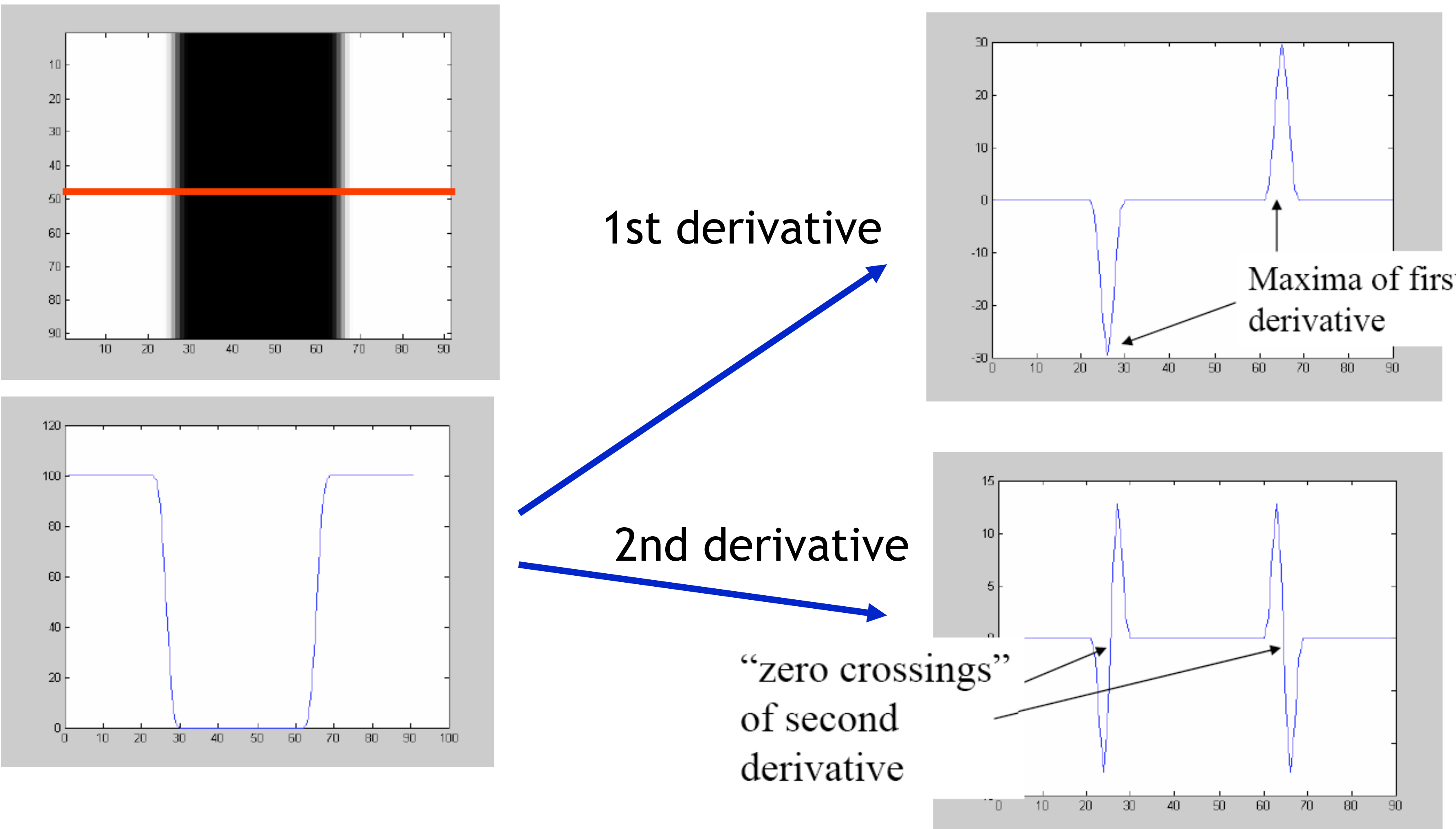
- Both, the BOX filter and the Gaussian filter are separable:
  - first convolve each row with a 1D filter
  - then convolve each column with a 1D filter

$$(f_x \otimes f_y) \otimes I = f_x \otimes (f_y \otimes I)$$

  - remember:
    - convolution is linear - associative and commutative
- Example: separable BOX filter

$$\underset{f_x \otimes f_y}{\begin{array}{|c|c|c|}\hline \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \hline \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \hline \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \hline \end{array}} = \underset{f_x}{\begin{array}{|c|c|c|}\hline \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \hline \end{array}} \otimes \underset{f_y}{\begin{array}{|c|}\hline \frac{1}{3} \\ \hline \frac{1}{3} \\ \hline \frac{1}{3} \\ \hline \end{array}}$$

Edges & Derivatives…



Compute Derivatives

$$\frac{d}{dx} f(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h} \approx f(x+1) - f(x)$$

- we can implement this as a linear filter:
  - direct:

-1	1

  - or symmetric:

-1	0	1

Edge-Detection

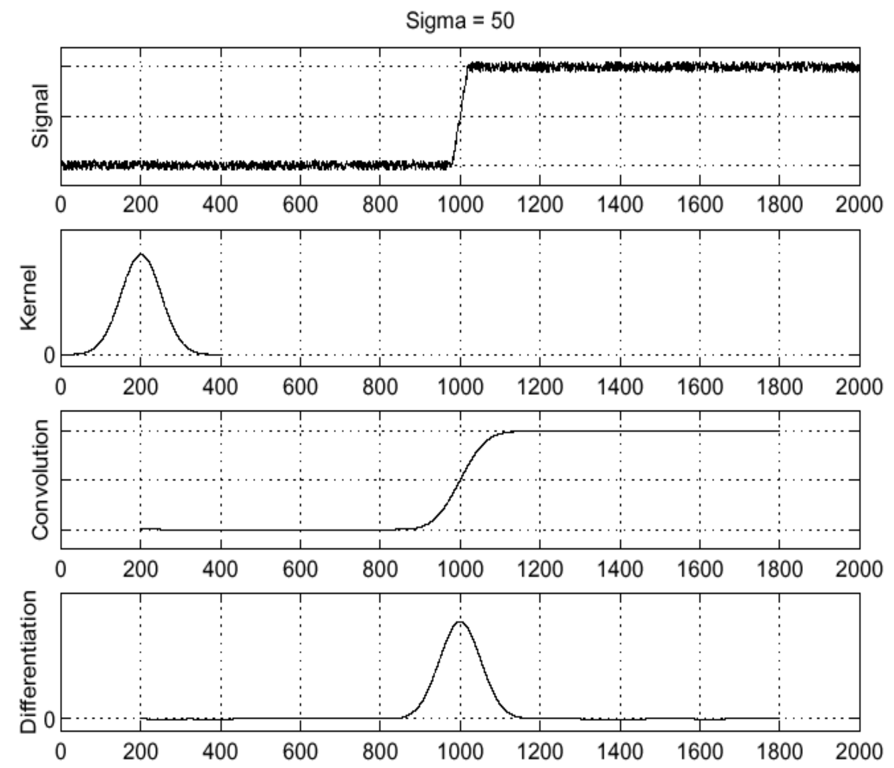
- based on 1st derivative:
    - smooth with Gaussian
    - calculate derivative
    - finds its maxima

$$f$$

$$g$$

$$g \otimes f$$

$$\frac{d}{dx}(g \otimes f)$$



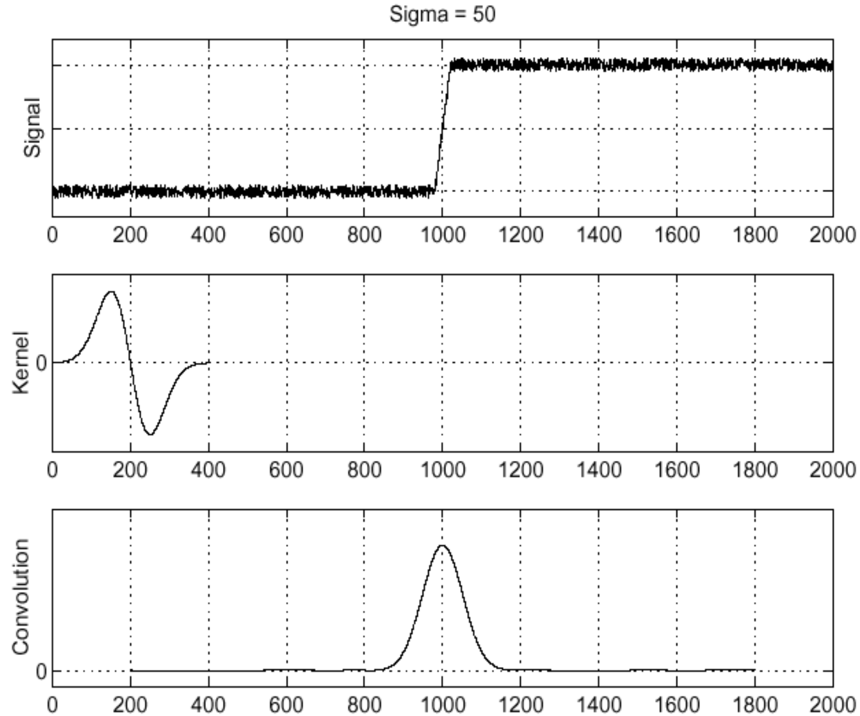
Edge-Detection

- Simplification: $$\frac{d}{dx}(g \otimes f) = \left(\frac{d}{dx}g\right) \otimes f$$
  - remember:
    derivative as well as convolution are linear operations
  - saves one operation

$$f$$

$$\frac{d}{dx}g$$

$$\left(\frac{d}{dx}g\right) \otimes f$$



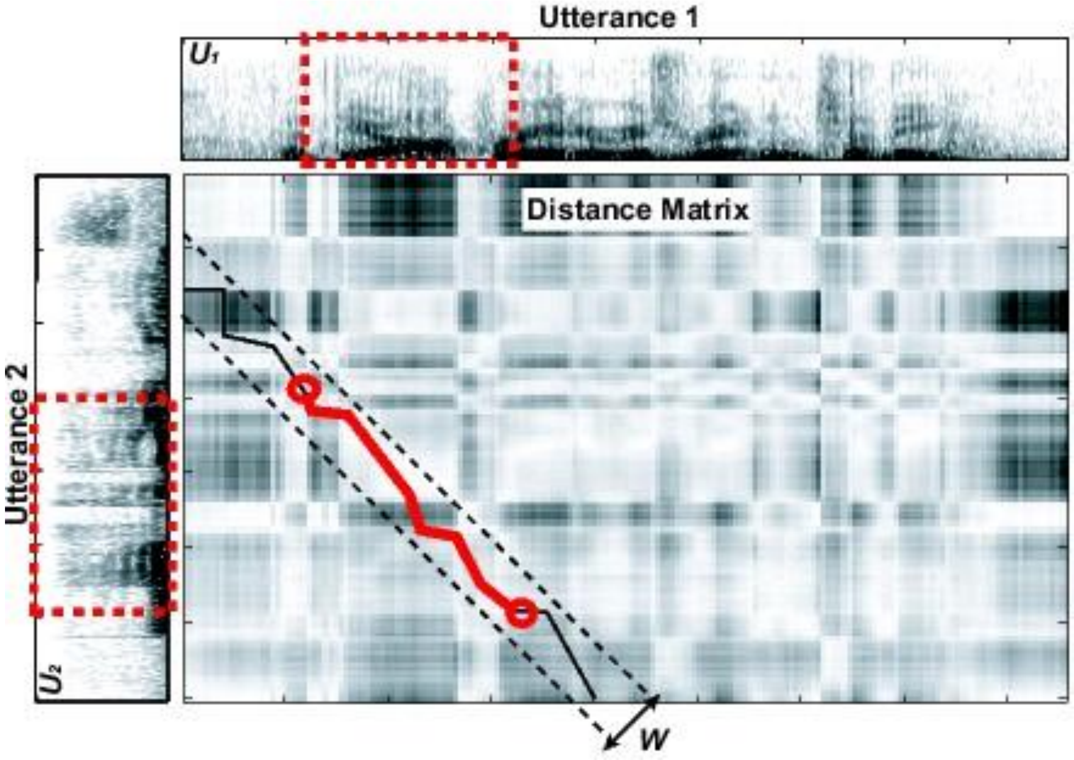
Luigi Cinque, Fabio Galasso and Marco Raoul Marini

Basics of
Automatic Speech Recognition (ASR)

Sapienza
Università di Roma

History of Automatic Speech Recognition

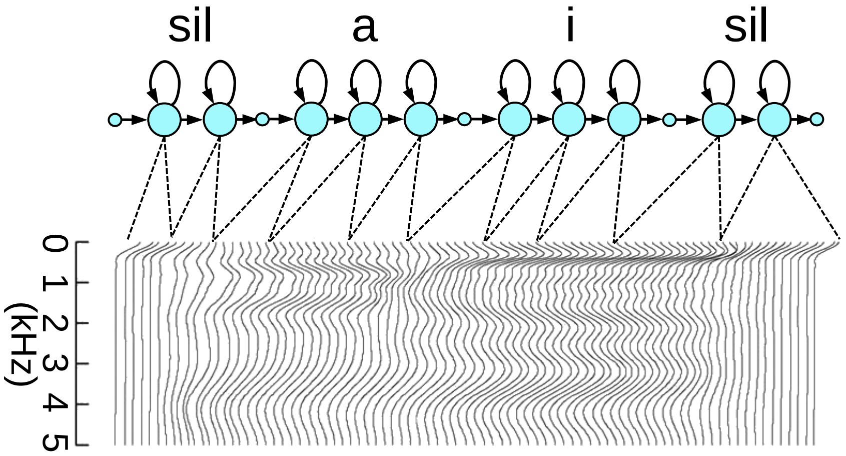
- Early 1970s: Dynamic Time Warping (DTW) to handle time variability
  - Distance measure for spectral variability



Source: http://publications.csail.mit.edu/abstracts/abstracts06/malex/seg_dtw.jpg

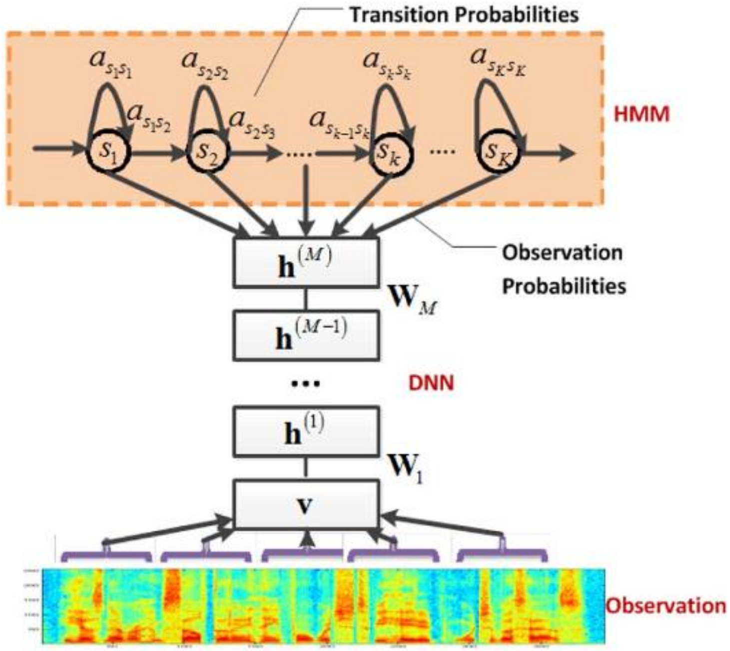
History of Automatic Speech Recognition

- Early 1970s: Dynamic Time Warping (DTW)
- Mid-Late 1970s: Hidden Markov Models (HMMs) become popular
  - statistical models of spectral variations, for discrete speech signals
- Mid 1980s: HMMs are the dominant technique for ASR



History of Automatic Speech Recognition

- Early 1970s: Dynamic Time Warping (DTW)
- Mid-Late 1970s: Hidden Markov Models (HMMs) become popular
- Mid 1980s: HMMs are the dominant technique for ASR
- 1990s: Large vocabularies and continuous dictation
- 2000s: Discriminative training (minimize word/phone error rate)
- 2010s: Deep learning significantly reduces error rate



George E. Dahl, et al. Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition. IEEE Trans. Audio, Speech & Language Processing, 2012.

Automatic Speech Recognition

- Find the most likely sentence (word sequence) $\boldsymbol{W}$, which transcribes the speech audio $\boldsymbol{A}$:

$$\widehat{\boldsymbol{W}} = \underset{\boldsymbol{W}}{\operatorname{argmax}}\, P(\boldsymbol{W}|\boldsymbol{A}) = \underset{\boldsymbol{W}}{\operatorname{argmax}}\, P(\boldsymbol{A}|\boldsymbol{W})P(\boldsymbol{W})$$

  - Acoustic model $P(\boldsymbol{A}|\boldsymbol{W})$
  - Language model $P(\boldsymbol{W})$

- Training: optimize the acoustic and language models separately
  - Speech Corpus: speech waveform and human-annotated transcriptions
  - Language model: with extra data (prefer daily expressions corpus for spontaneous speech)

Basics of Text Analysis and Natural Language Processing (NLP)

Sapienza
Università di Roma

Word Similarity

- Task: given two words, predict how similar they are

- The Distributional Hypothesis:

You shall know a word by the company it keeps

(John Firth, 1957)

Distributional Hypothesis (J.R. Firth 1957)

- Words that occur in **similar contexts** tend to have **similar meanings**
  - “You shall know a word by the company it keeps”
  - “If A and B have almost identical environments ”

- Words which are **synonyms** tend to occur in the **similar context**

Intuition of distributional word similarity

- Suppose I asked you what is **tesgüino**?

A bottle of **tesgüino** is on the table
Everybody likes **tesgüino**
Tesgüino makes you drunk

- From context words, humans can guess **tesgüino** an alcoholic beverage like beer
- Intuition for the algorithm:
  - Two words are similar if they have **similar** word **contexts**

🌟 Vector semantics

- **Goal**: Learning **representations** (embeddings) of the meaning of words, directly from their **distributions** in text
- Important for NLP applications that make use of meaning
  - Question Answering, Summarization, Detecting paraphrases or plagiarism and dialogue

Luigi Cinque, Fabio Galasso and Marco Raoul Marini

Term-document matrix

- Count of word w in a document d:
  - Each document in a count vector in $\mathbb{N}^{V}$

Document

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

Word / Term

Word-word matrix

- Instead of an entire document, use smaller contexts
  - Paragraph
  - Window of +-4 words

- Word is now defined by counts of context words

	aardvark	...	computer	data	result	pie	sugar	...
cherry	0	...	2	8	9	442	25	
strawberry	0	...	0	0	1	60	19	
digital	0	...	1670	1683	85	5	4	
information	0	...	3325	3982	378	5	13	

TF-IDF: Weighting terms in the vector

- Not all words are equally important
  - Some words just co-occur frequently with many different words (e.g. the, they, it)
- **Term Frequency:** Words that occur nearby frequently (maybe *pie* nearby *cherry*) are more important.
- **Inverse Document Frequency:** Words that are too frequent may be unimportant (e.g. *the*, *it, he*, she).

Luigi Cinque, Fabio Galasso and Marco Raoul Marini

Measuring similarity

- We have words *w* and *v*
- How do we measure their similarity?

- **Dot product** or **inner product** from linear algebra

$$dot\text{-}product(\vec{v},\vec{w}) = \vec{v}\cdot\vec{w} = \sum_{i=1}^{N} v_i w_i$$

  - High when two vectors have large values in same dimensions
  - Low (actually 0) for **orthogonal vectors**

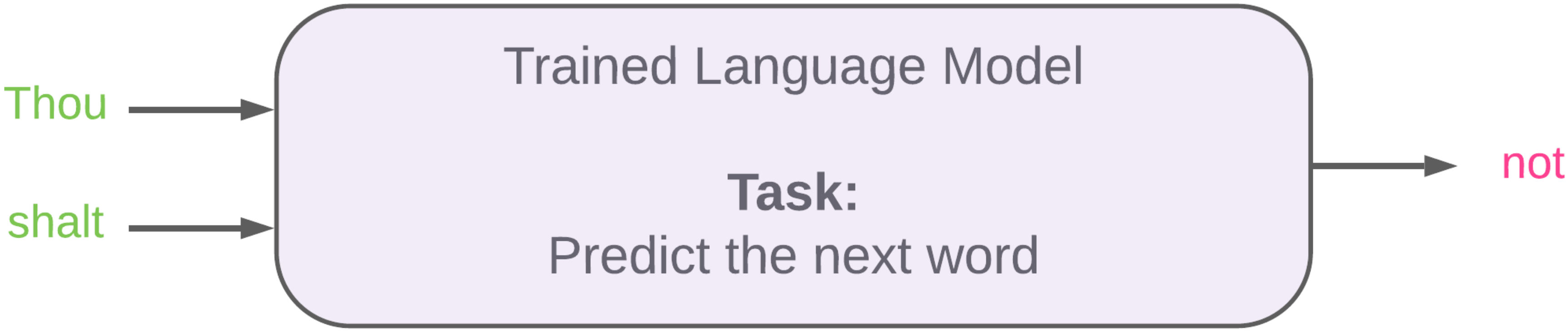
Review: Language Modeling

- Next-word prediction task

input # 1 input # 2 output

Thou shalt ______

Review: Language Modeling



Review: Language Modeling

Thou shalt not make a machine in the likeness of a human mind

Sliding window across text

thou	shalt	not	make	a	machine	....

Dataset

input 1	input 2	output
thou	shalt	not

Thou shalt not make a machine in the likeness of a human mind

Sliding window across text

thou	shalt	not	make	a	machine	....
thou	shalt	not	make	a	machine	....

Dataset

input 1	input 2	output
thou	shalt	not
shalt	not	make

Review: Language Modeling

Thou shalt not $\boxed{\text{make a machine}}$ in the likeness of a human mind

Sliding window across text

thou	shalt	not	make	a	machine	....
thou	shalt	not	make	a	machine	....
thou	shalt	not	make	a	machine	....
thou	shalt	not	make	a	machine	....

Dataset

input 1	input 2	output
thou	shalt	not
shalt	not	make
not	make	a
make	a	machine

Thank you

Acknowledges: some slides and material from Bernt Schiele, Mario Fritz, Michael Black, Bill Freeman, Fei-Fei, Justin Johnson, Serena Yeung, Yining Chen, Anand Avati, Andrew Ng

SAPIENZA
UNIVERSITÀ DI ROMA