Course Section 1

Xiao-Wei CAO
acdoge.cao@gmail.com

What happened in this week

Huixiang Chen
Pro. Ph.D. Student
in University of Florida.
R , I , P

The Hidden Story Behind the Suicide PhD Candidate Huixiang Chen

Huixiang Voice Follow
Jun 29 · 5 min read

At June 13 2019, just before the ISCA 2019 conference in Phoenix, a doctoral candidate, whose paper was supposed to be published in this conference, hanged himself in the campus building of University of Florida.

3D-based Video Recognition Acceleration by Leveraging Temporal Locality

Huixiang Chen[*], Mingcong Song[*], Jiechen Zhao[*], Yuting Dai[‡], Tao Li[*]
[*]IDEAL Lab, University of Florida; [†]Guizhou University
{stanley.chen, songmingcong, jiechen.zhao}@ufl.edu, yutingdai90@gmail.com, taoli@ece.ufl.edu

Published in:

· Proceeding
ISCA '19 Proceedings of the 46th International Symposium on Computer Architecture
Pages 79-90

Phoenix, Arizona — June 22 - 26, 2019
ACM New York, NY, USA ©2019
table of contents ISBN: 978-1-4503-6669-4 doi>10.1145/3307650.3322260

- The Hidden Story Behind the Suicide PhD Candidate Huixiang Chen – Medium
- 一位博士生选择自杀，在论文中了顶会之后 - 量子位
- 如何看待佛罗里达大学 ECE 博士生陈慧祥自杀？ - 知乎

What happened in this week

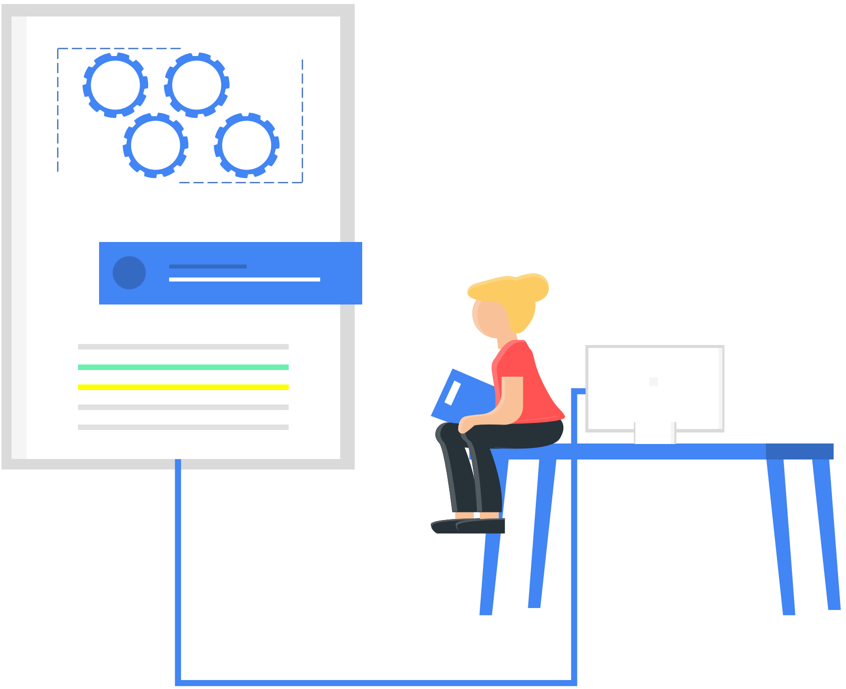
ddbourgin Update README.md	
gmm	add axis argument to logsumexp
hmm	remove scipy import from testing
lda	Update README.md
linear_models	add ridge regression
neural_nets	Update README.md
ngram	remove filter_punctuation from word tokenization
nonparametric	add nonparametric module
preprocessing	update readme
rl_models	add dyna plot
trees	Update README.md
utils	update readme
.gitignore	ignore TODO file
LICENSE	add license
README.md	update readme

David Bourgin
[GitHub]

- 惊为天人，NumPy手写全部主流机器学习模型，代码超3万行 – 机器之心

Contents

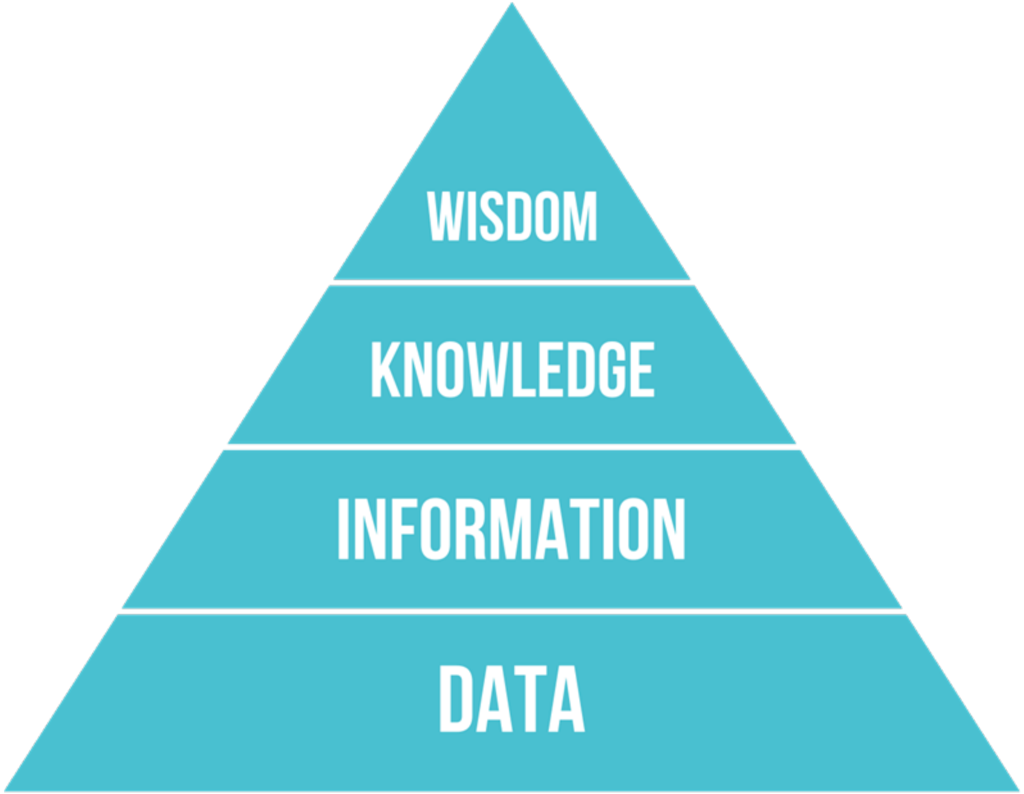
- Quick Review
  - Data, Information and Knowledge
  - Machine Learning Basic Concepts
  - Data in Machine
- Quiz Discussion
- Supplement
  - Digital Image Processing – Filter
  - Decision Tree
- Warm Up for Next Week
  - Linear Model
  - Support Vector Machine



Data, Information and Knowledge

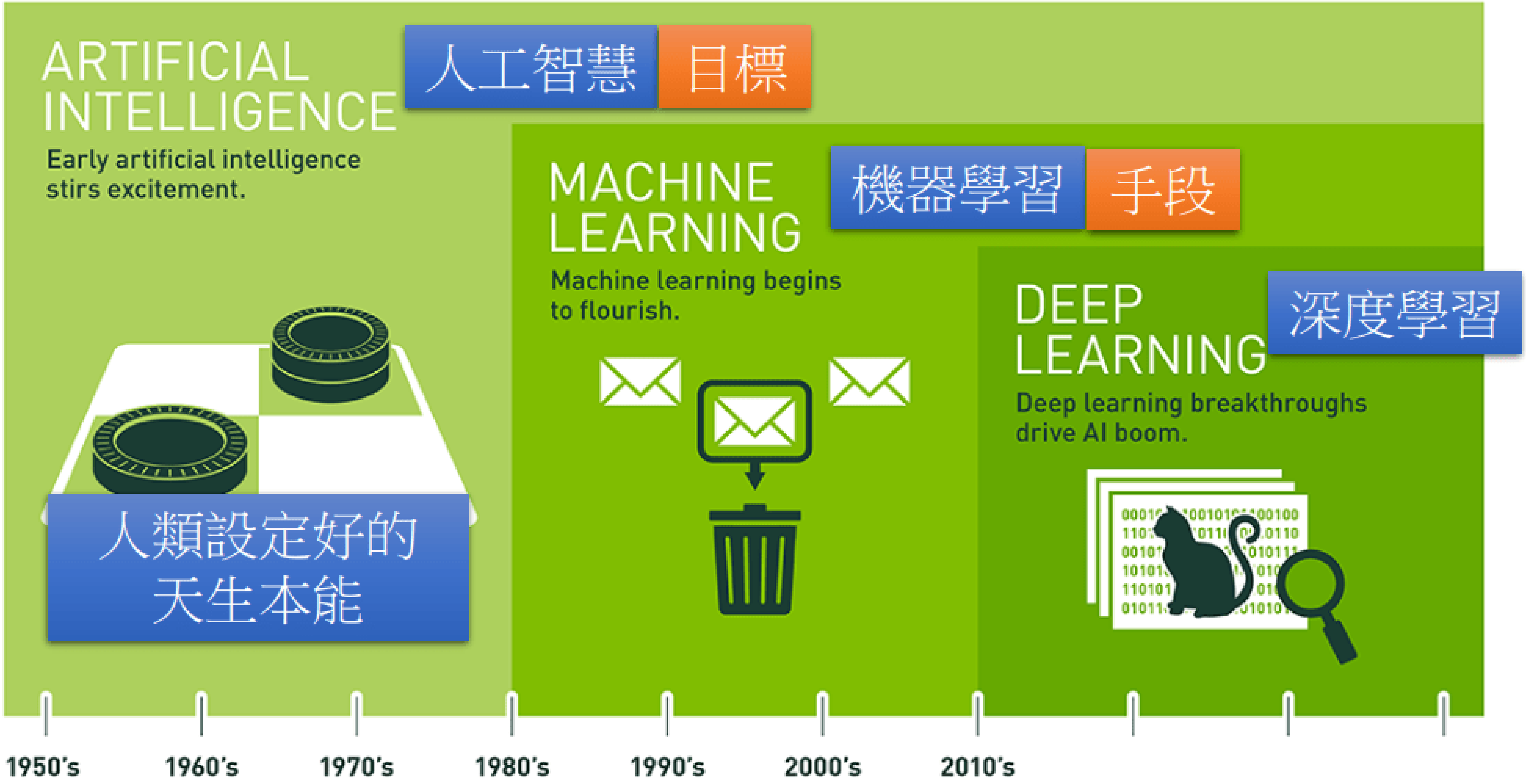
- Definition: …
- How humans collect data
- How humans gain information
- How humans learn knowledge
- Relationship

Human → Machine



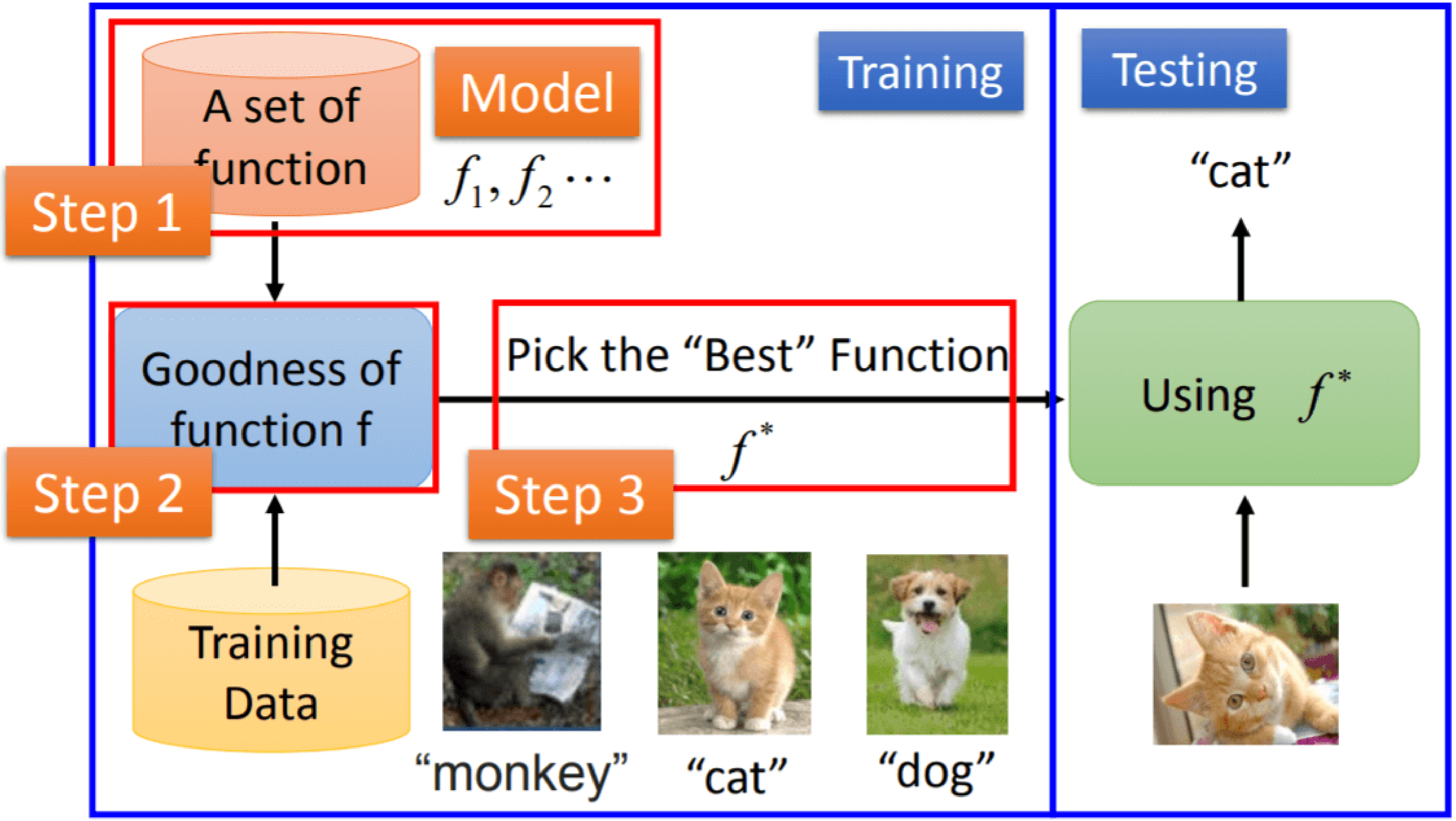
- Difference between data, information and knowledge? – Medium
- DIKW pyramid – Wikipedia
- Data, Information, and Knowledge in Visualization - IEEE

AI / ML / DL



https://blogs.nvidia.com.tw/2016/07/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/

Machine Learning Basic Concepts

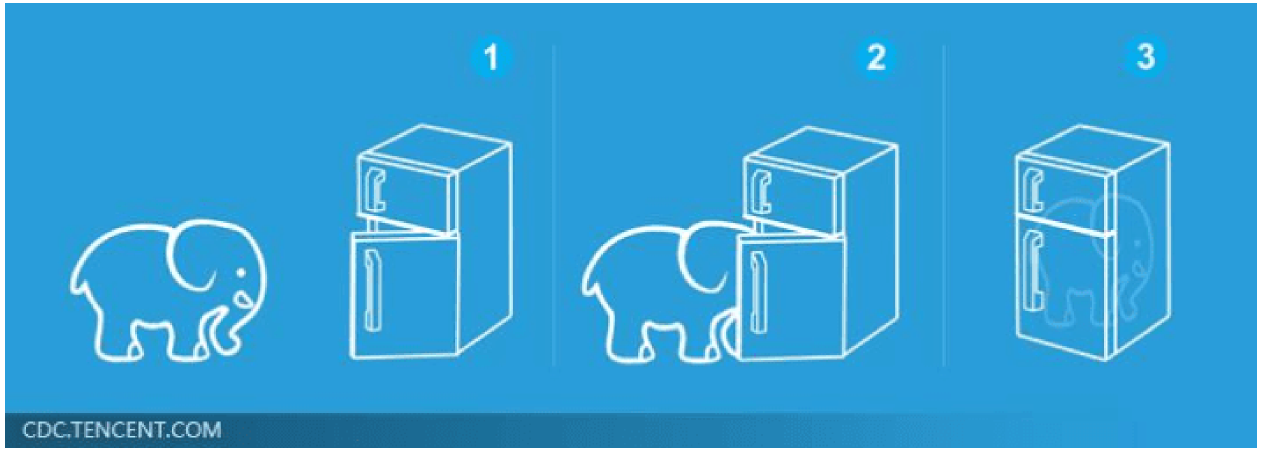


http://speech.ee.ntu.edu.tw/~tlkagk/courses_ML17_2.html

Machine Learning is Simple

Step 1: define a set of function → Step 2: goodness of function → Step 3: pick the best function

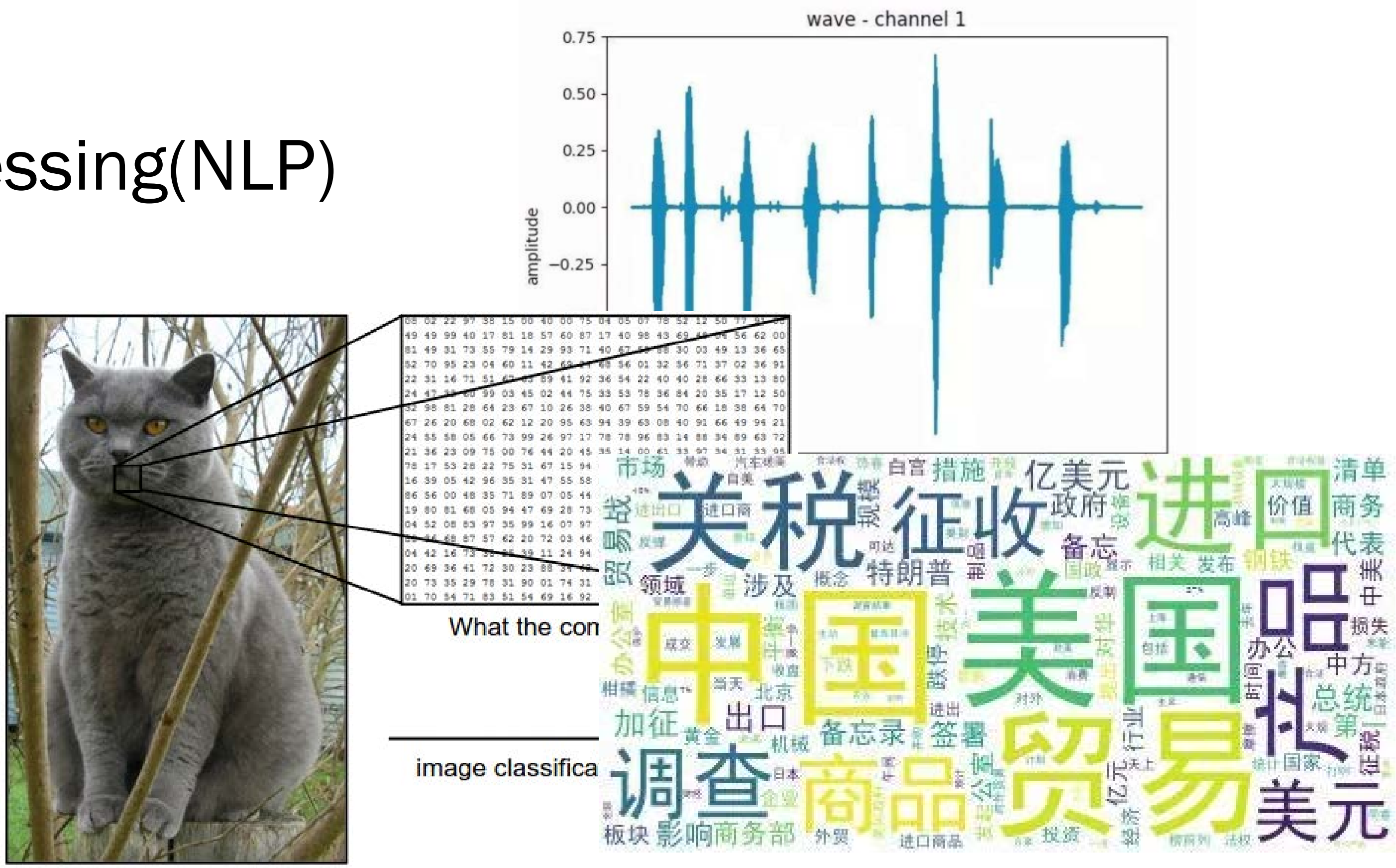
就好像把大象放進冰箱 ……



http://speech.ee.ntu.edu.tw/~tlkagk/courses_ML17_2.html

Data in Machine

- Computer Vision(CV)
  - Image / Video
- Nature Language Processing(NLP)
- Speech Recognition
- More …



https://cs231n.github.io/classification/

From Network

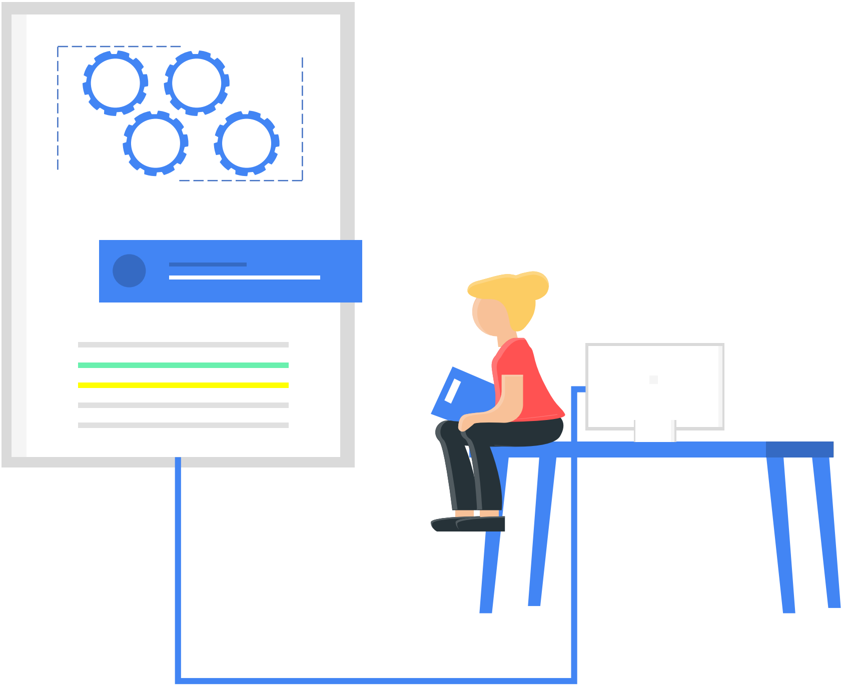
A Sad Real Story...

Day 1

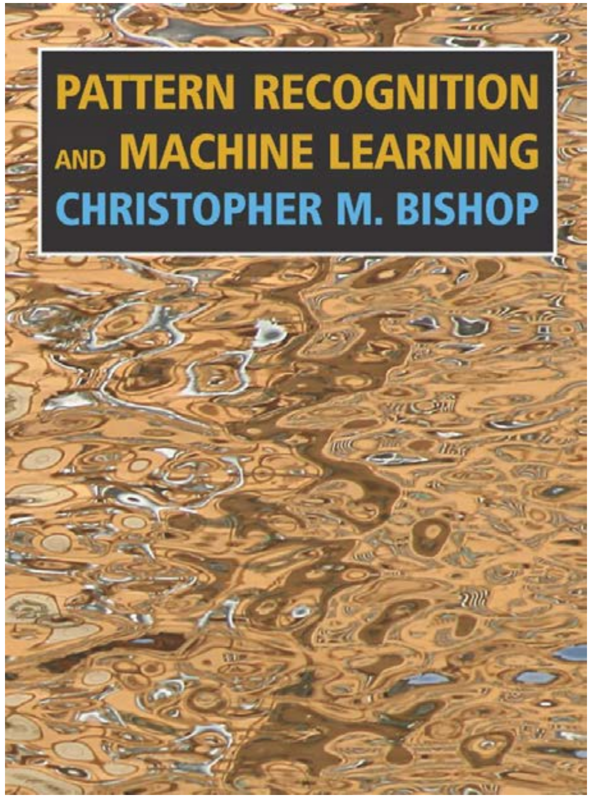
Day 2

Day 3

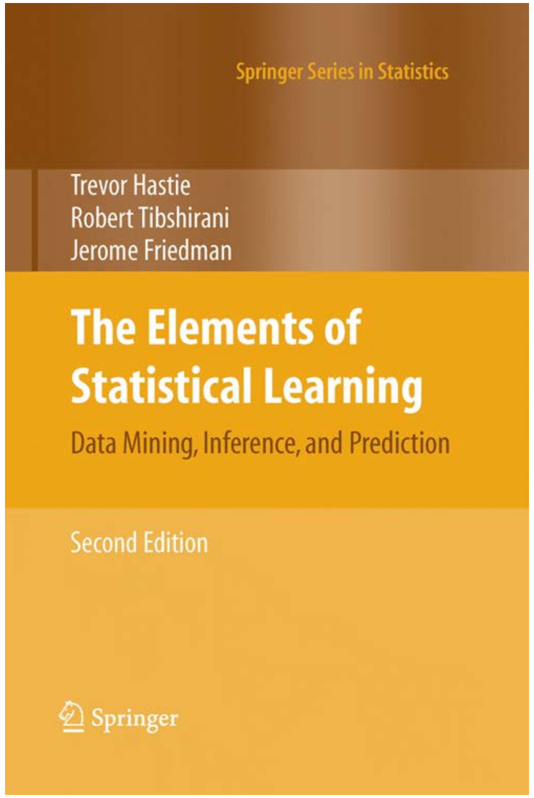
Contents



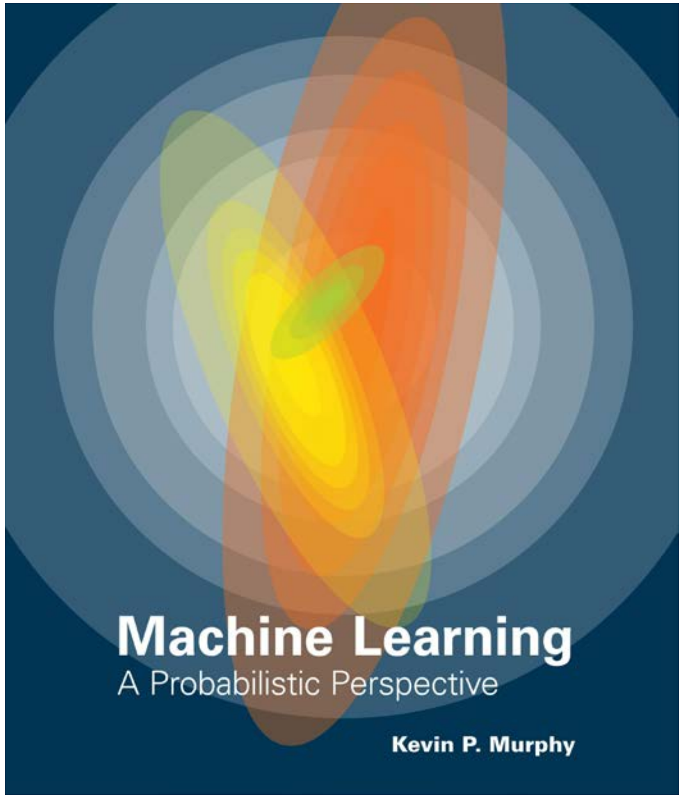
Textbook



PRML



ESL



MLAPP

“My English is poor” ☹

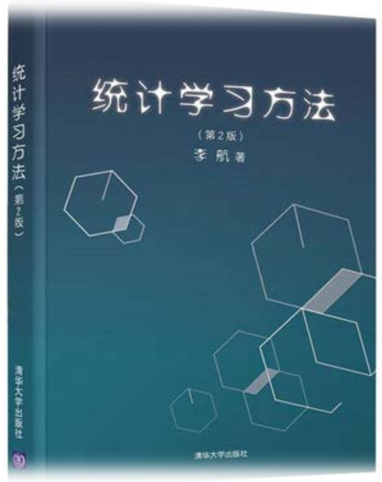
Chinese Textbook

西瓜书 [如何使用本书] [勘误修订] [公式推导(非官方)]

- 首先，读者诸君务须注意，本书是一本教科书.
- 第二，这是一本入门级教科书.
- 第三，这是一本面向理工科高年级本科生和研究生的教科书.
- 第四，这本书不妨多读几遍.

- 本书是一部机器学习的基本读物
- 要求读者拥有高等数学、线性代数和概率统计基础
- 主要讲述统计机器学习
- 给出细致的数学推导和具体实例

注：这本书的第二版 2019 年 3 月出版



How to Read Books for Research

- Choose the Right Books. (Understand your current level)
- Read the Book, Don't Scan It.
- Take Breaks Between Chapters.
- Finish and Reflect.
- Organize What You've Learned.
- Coming Back Later.
- Wrapping Up.

www.ryanglover.net/articles/how-to-read-books-for-research

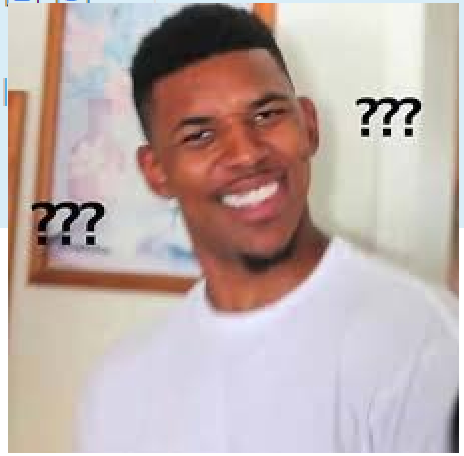
But if you are a beginner…

- Build consciousness first.
- Use analogy. (From figuration to abstraction)
- Think subject association.
- Thanks to the Internet
  - Search Engine - Google
  - Blog – CSDN / CNBLOGS
  - Forum – Like our course
  - MOOC – Andrew Ng
  - Open Course – CS229

看山是山
看水是水

Learning How to Learn

课外	-	**能力初探，编程环境准备** 接触到新的名词概念时应该怎么办? 如何有效使用谷歌搜索并解决问题? 如何安装 Python 科学计算环境? 不同的 Python 环境如何进行管理? 如何搭建交互式 Python 编程环境? 如何对自己的代码进行版本控制? 哪里可以找到开源项目代码? 如何参与 GitHub 组织协作? [学习的十大好习惯和坏习惯]	**官方网站:** Wikipedia Google Zhihu Bilibili Anaconda VS Code Python Numpy Pandas Matplotlib Scikit-learn Jupyter Notebook Git GitHub Bitbucket **参考资料:** 搜索引擎有哪些常用技巧? A Byte of Python [译] *进阶 [1] [2] [3] Morvan - Python 数据处理 Git CS 231n Python Numpy Tutorial 利用 Python 进行数据分析 Pro Git [译] Github 简明教程	Python 2.7 于 2020 年不再提供维护，建议使用 3.5+ 版本。 请预览"预备知识"中数学内容快速回顾，并完整阅读底部 FAQ.

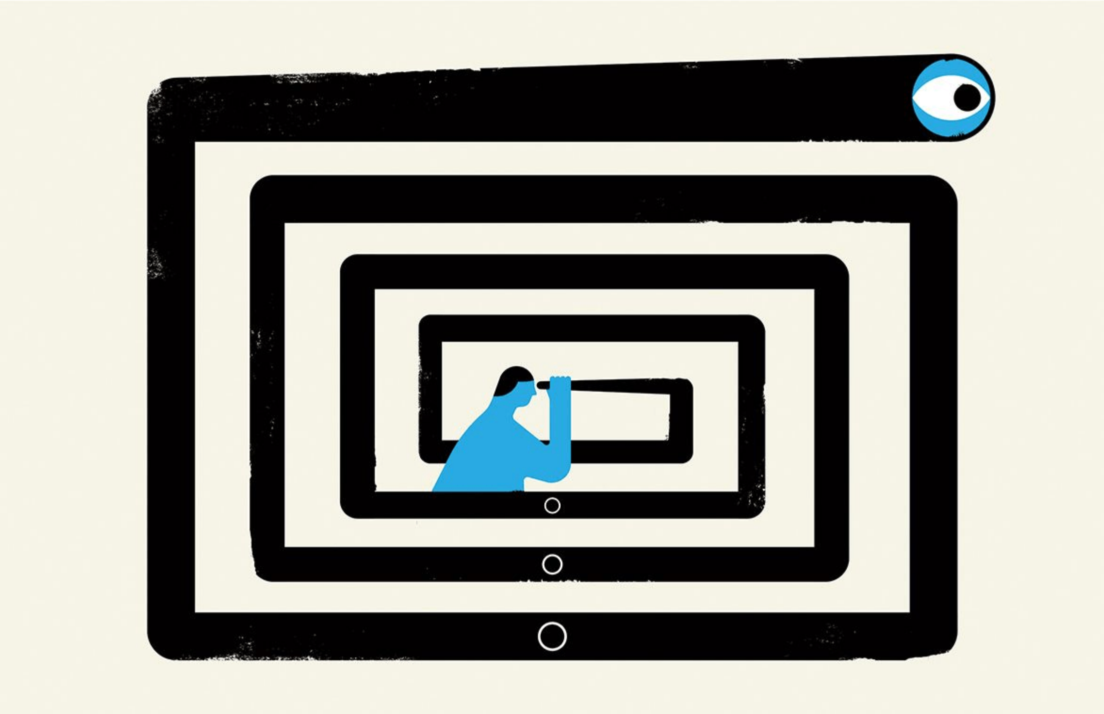


"Be curious. Read widely. Try new things. What people call intelligence just boils down to **curiosity**." -- Aaron Swartz

"Meta Learning" in deep learning.

Why use Jupyter Notebook

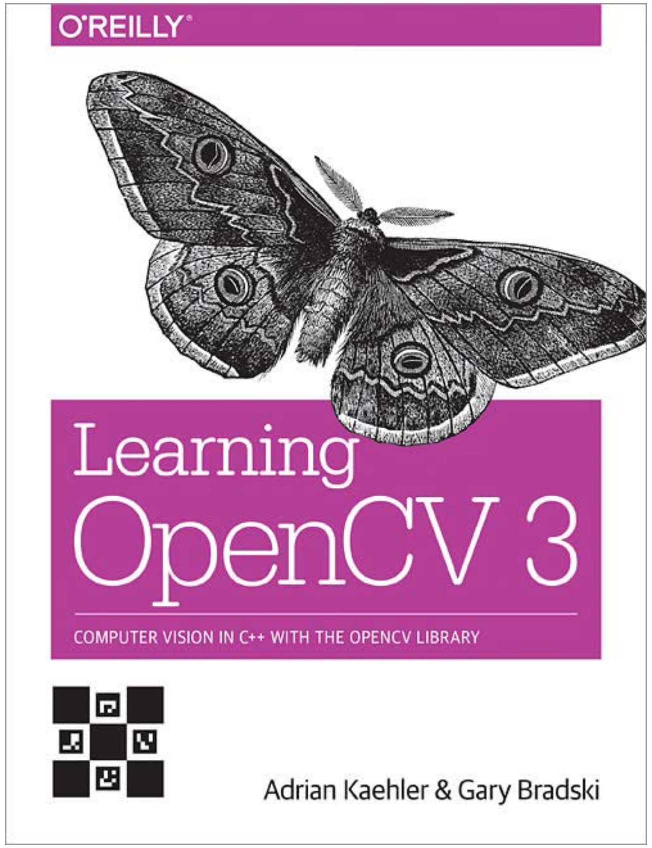
- Data exploration
- Speak my language
- Problems noted



- Why Jupyter is data scientists' computational notebook of choice - Nature

*Problems met in Notebooks

- Python Slice Syntax / Axis in Numpy
- Color Histogram
- HOG / LBP / Haar
- More OpenCV projects [GitHub]



Practice the Theory you learned by tools.

We’ll use IDE later

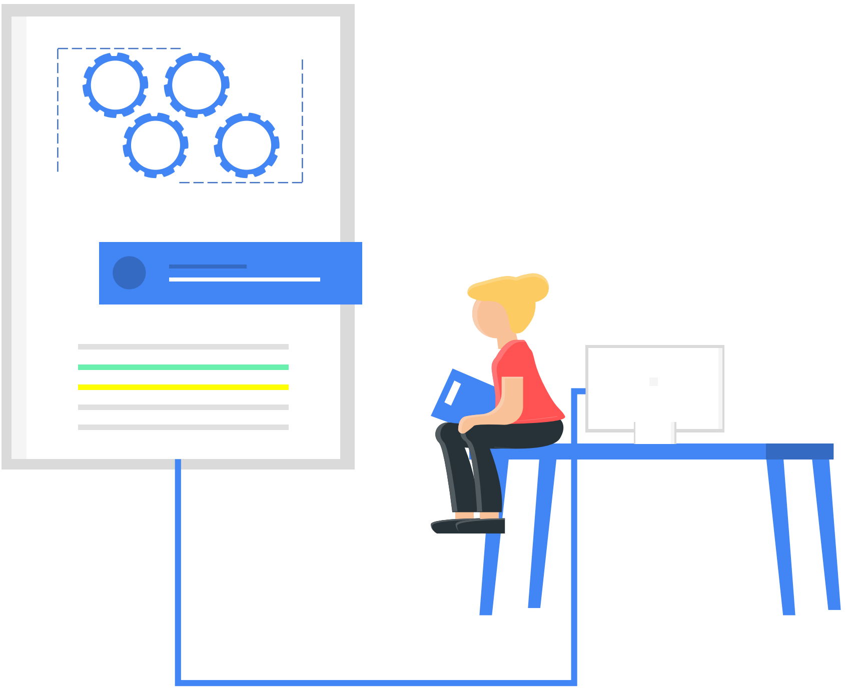
- Jetbrains Pycharm Professional [Website]
  - Free individual licenses for students
  - Using our **edu.cn** e-mail
- For Remote Developing
  - Connect to lab’s serve
  - Using GPU to accelerate
  - You can still code locally

Break

5 mins

Contents

- Quick Review
  - Data, Information and Knowledge
  - Machine Learning Basic Concepts
  - Data in Machine
- Quiz Discussion
- Supplement
  - Digital Image Processing - Filter
  - Decision Tree
- Warm Up for Next Week
  - Linear Model
  - Support Vector Machine



Digital Image Processing

- Note: Most of the following content comes from
  - Stanford EE368 course Digital Image Processing [Main Lecture Notes].
  - Stanford CS231a Lecture 3: Linear Filters [VISION Lab Teach]

Bernd Girod

Fei-Fei Li

Linear Image Processing

- Image processing system $S(.)$ is linear, iff superposition principle holds:

$$S(\alpha \cdot f[x,y] + \beta \cdot g[x,y]) = \alpha \cdot S(f[x,y]) + \beta \cdot S(g[x,y]) \text{ for all } \alpha, \beta \in \mathbb{R}$$

- Any linear image processing system can be written as

$$\vec{g} = H\vec{f}$$

Note: matrix H need not be square.

- by sorting pixels into a column vector

$$\vec{f} = (f[0,0] \quad f[1,0] \quad \cdots \quad f[N-1,0] \quad f[0,1] \ldots f[N-1,1] \ldots \cdots f[0,L-1] \cdots f[N-1,L-1])^T$$

Impulse response

- Another way to represent any linear image processing scheme

$$g[\alpha,\beta] = \sum_{x=0}^{N-1}\sum_{y=0}^{L-1} \delta[x-a, y-b]\cdot h[x,\alpha,y,\beta] = h[a,\alpha,b,\beta]$$

impulse response

- Input: unit impulse at pixel $[a, b]$

$$f[x,y] = \delta[x-a,y-b] = \begin{cases} 1 & x=a \wedge y=b \\ 0 & \text{else} \end{cases}$$

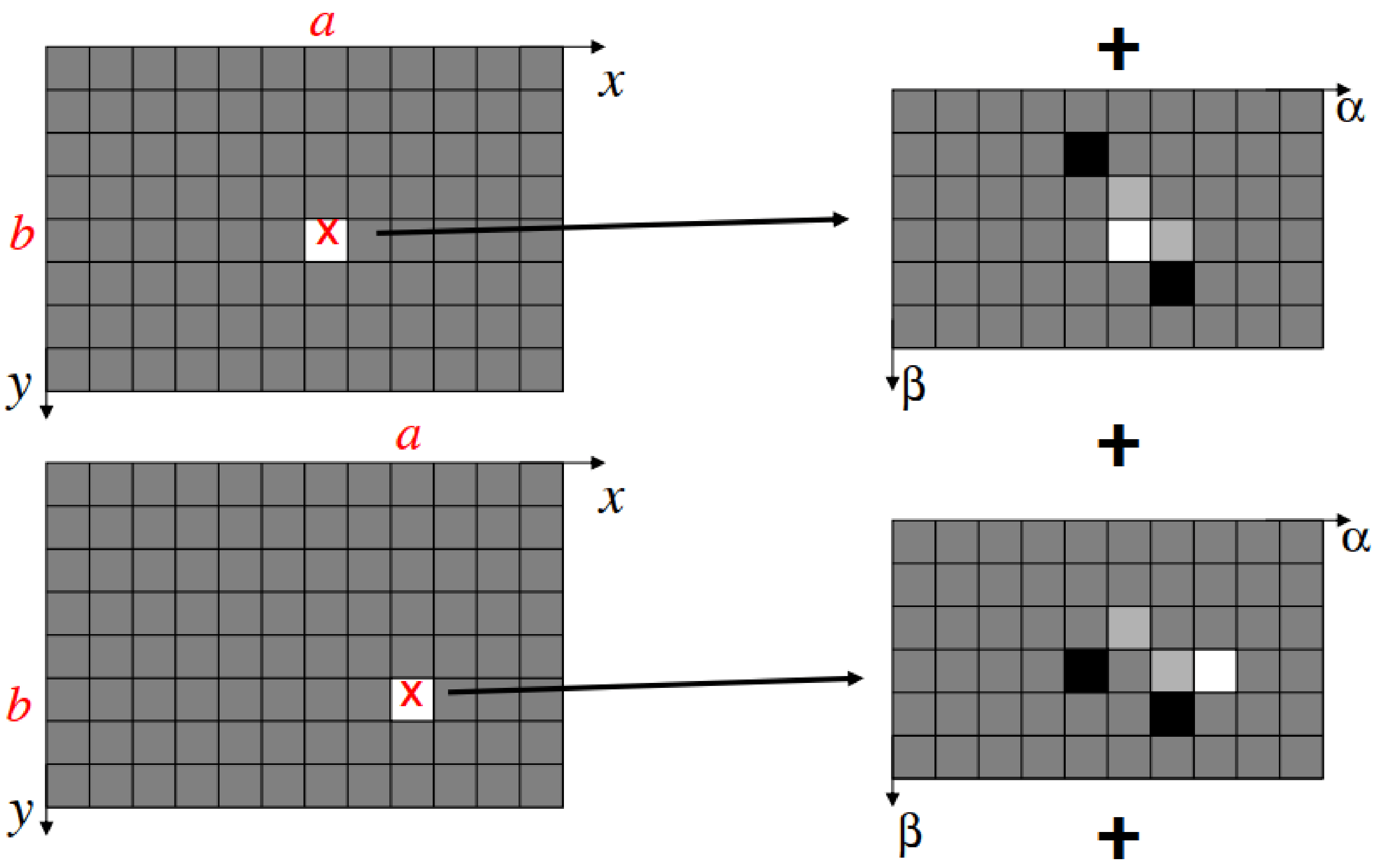
- Output: impulse response

$$g[\alpha,\beta] = \sum_{x=0}^{N-1}\sum_{y=0}^{L-1} f[x,y]\cdot h[x,\alpha,y,\beta]$$

Interpretation #1: superposition of impulse responses

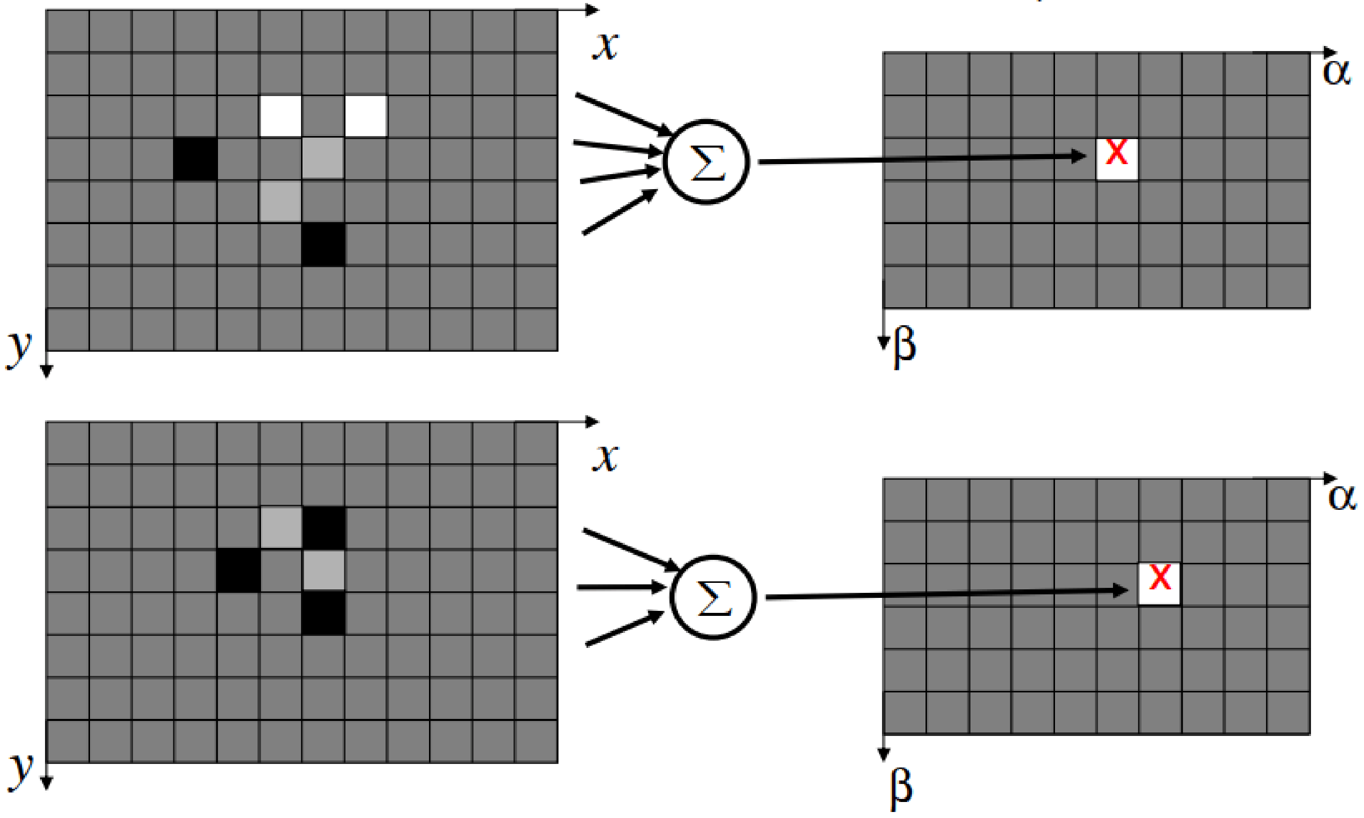
$\delta[x-a,y-b]f[a,b]$

$g[\alpha,\beta]=h[a,\alpha,b,\beta]f[a,b]$

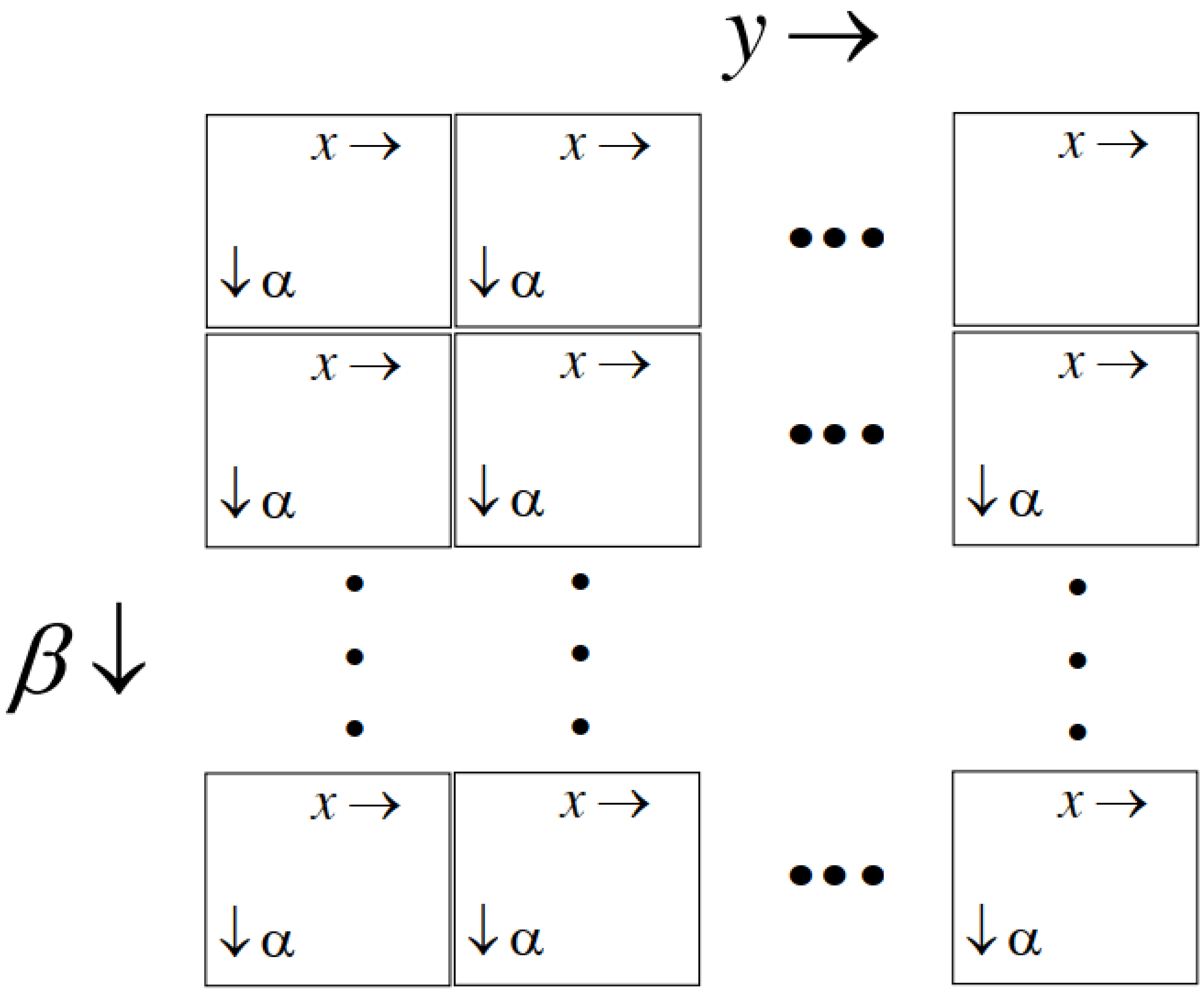


Interpretation #2: linear combination of input values

$$f[x,y]\cdot h[x,\alpha,y,\beta] \qquad g[\alpha,\beta]=\sum_{x=0}^{N-1}\sum_{y=0}^{L-1} f[x,y]\cdot h[x,\alpha,y,\beta]$$



Relationship of H and $h[x, \alpha, y, \beta\ ]$



Separable linear image processing

- Impulse response is separable in $[x, \alpha]$ and $[y, \beta]$, i.e., can be written as

$$g[\alpha, \beta] = \sum_{x=0}^{N-1} \sum_{y=0}^{L-1} f[x, y] \cdot h_x[x, \alpha] h_y[y, \beta]$$

- Processing can be carried out
  - row by row, then column by column

$$g[\alpha, \beta] = \sum_{y=0}^{L-1} h_y[y, \beta] \sum_{x=0}^{N-1} f[x, y] \cdot h_x[x, \alpha]$$

  - column by column, then row by row

$$g[\alpha, \beta] = \sum_{x=0}^{N-1} h_x[x, \alpha] \sum_{y=0}^{L-1} f[x, y] \cdot h_y[y, \beta]$$

Separable linear image processing (cont.)

- If the digital input and output images are written as a matrices f and g, we can conveniently write

$$\mathbf{g} = \mathbf{H_y^T} \cdot \mathbf{f} \cdot \mathbf{H_x}$$

$$\mathbf{H}_y = \begin{bmatrix} h_y[0,0] & h_y[0,1] & \dots & h_y\left[0, L_g-1\right] \\ h_y[1,0] & h_y[1,1] & \dots & h_y\left[1, L_g-1\right] \\ \vdots & \vdots & & \vdots \\ h_y[L-1,0] & h_y[L-1,1] & \dots & h_y\left[L-1, L_g-1\right] \end{bmatrix} \quad \mathbf{H}_x = \begin{bmatrix} h_x[0,0] & h_x[0,1] & \dots & h_x\left[0, N_g-1\right] \\ h_x[1,0] & h_x[1,1] & \dots & h_x\left[1, N_g-1\right] \\ \vdots & \vdots & & \vdots \\ h_x[N-1,0] & h_x[N-1,1] & \dots & h_x\left[N-1, N_g-1\right] \end{bmatrix}$$

- Output image **g** has size $L_g \times N_g$
- If the operator does not change image size, H_x and H_y are square matrices

Example: filtering

- Each pixel is replaced by the average of two horizontally (vertically) neighboring pixels

$$\mathbf{H}_{\mathbf{x}} = \mathbf{H}_{\mathbf{y}} = \begin{bmatrix} 0.5 & 0 & & \cdots & & & & 0 \\ 0.5 & 0.5 & & & & & & \\ 0 & 0.5 & 0.5 & & & & & \\ & & 0.5 & 0.5 & \ddots & & & \vdots \\ & & & 0.5 & 0.5 & & & \\ \vdots & & & \ddots & 0.5 & 0.5 & & \\ & & & & & 0.5 & 0.5 & 0 \\ 0 & & & \cdots & & 0 & 0.5 & 0.5 \end{bmatrix}$$

- Shift-invariant operation (except for image boundary)

Shift-invariant systems and Toeplitz matrices

- For a separable, shift-invariant, linear system

$$\left[h_x[x,\alpha] = h_{siv/x}[\alpha - x]\right] \qquad h_y[y,\beta] = h_{siv/y}[\beta - y]$$

- Matrices H_x and H_y are square, and Toeplitz matrices, e.g.,

$$\mathbf{H_x} = \begin{bmatrix} h_{siv/x}[0] & h_{siv/x}[1] & \cdots & h_{siv/x}[N-1] \\ h_{siv/x}[-1] & h_{siv/x}[0] & \cdots & h_{siv/x}[N-2] \\ \vdots & \vdots & & \vdots \\ h_{siv/x}[1-N] & h_{siv/x}[2-N] & \cdots & h_{siv/x}[0] \end{bmatrix}$$

- Operation is a 2-d separable convolution („filtering“)

$$g[\alpha,\beta] = \sum_{x=0}^{N-1} \sum_{y=0}^{L-1} f[x,y] \times h_{\mathrm{siv}/x}[\alpha - x] h_{\mathrm{siv}/y}[\beta - y]$$

Non-separable 2-d convolution

- Convolution kernel of linear shift-invariant system („filter“) can also be non-separable

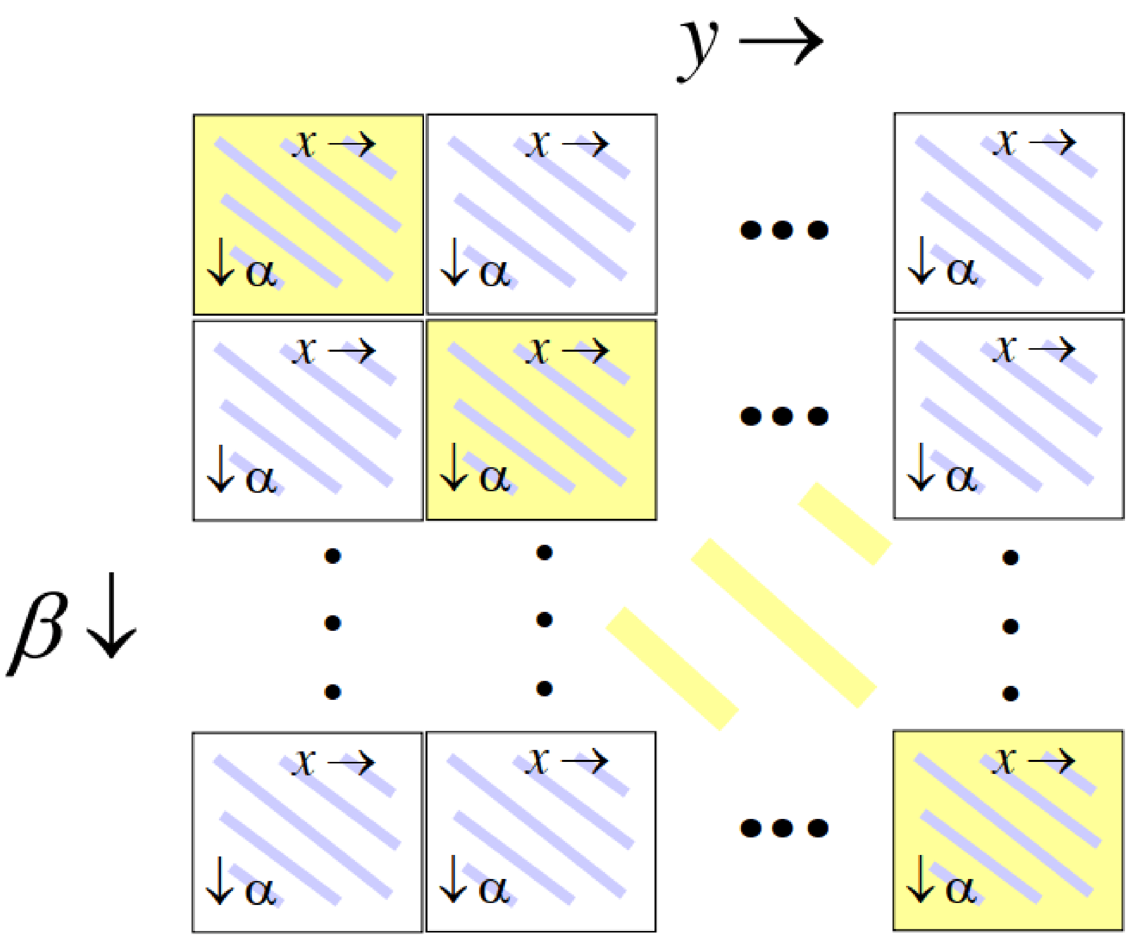
$$g[\alpha,\beta]=\sum_{x=0}^{N-1}\sum_{y=0}^{L-1} f[x,y]\cdot h_{siv}[\alpha-x,\beta-y]$$

- Viewed as a matrix operation . . .

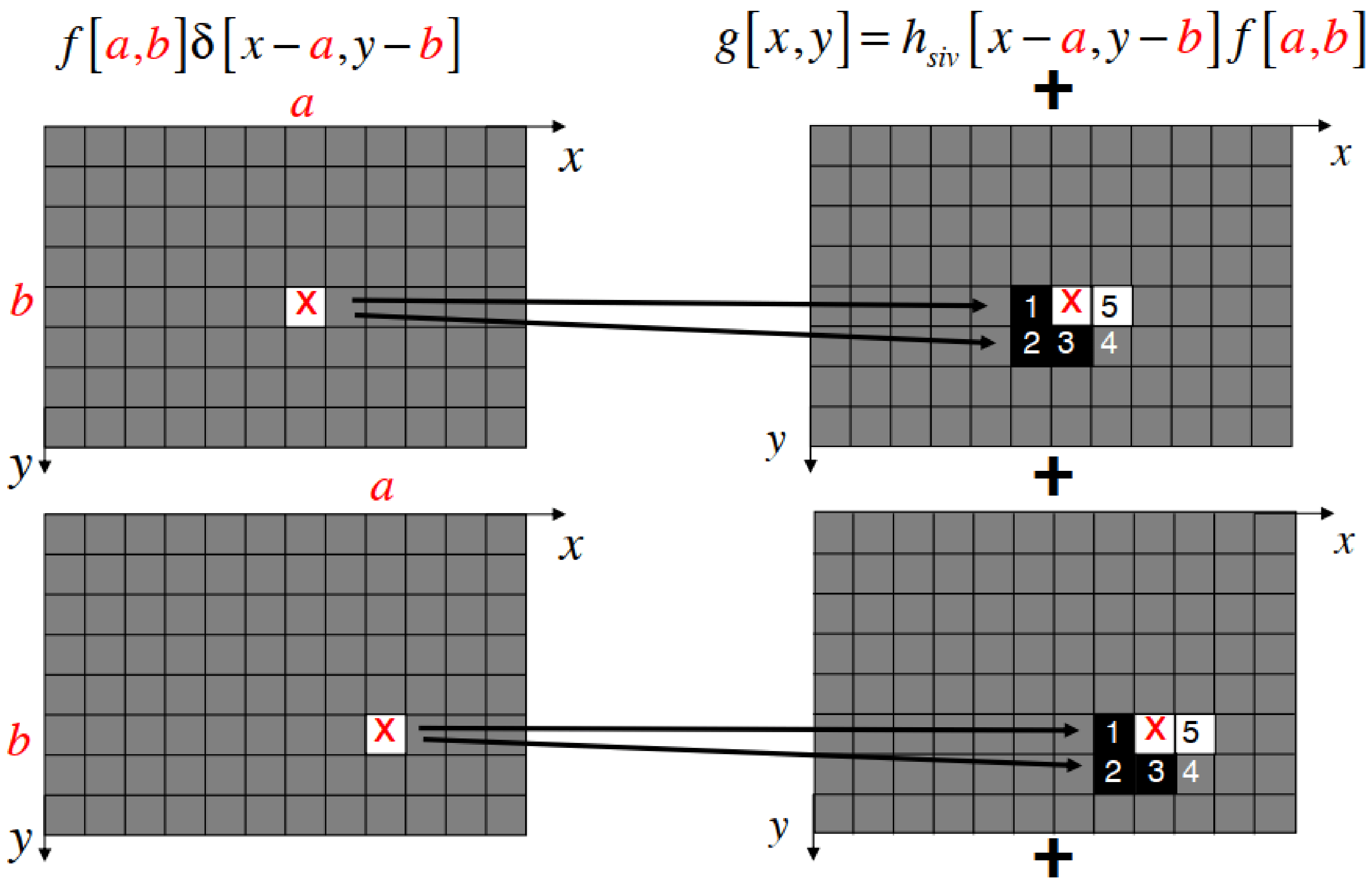
$$\vec{g}=H\vec{f}$$

. . . ***H*** is a block Toeplitz matrix

Structure of *H* for non-separable convolution

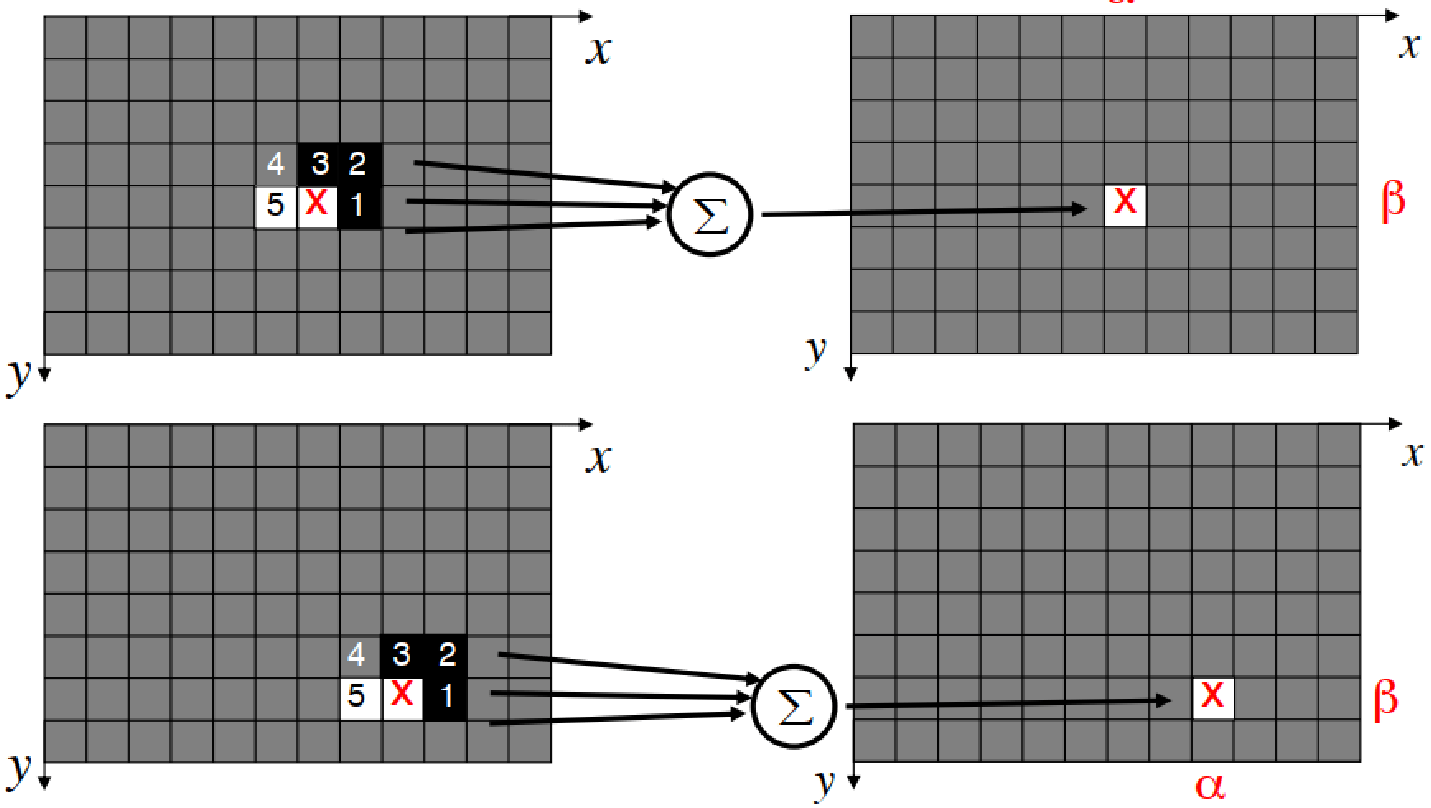


Convolution: superposition of impulse responses



Convolution: linear combination of neighboring pixel values

$$f[x,y]\cdot h_{siv}[\alpha - x, \beta - y] \qquad g[\alpha,\beta] = \sum_{x=0}^{N-1}\sum_{y=0}^{L-1} f[x,y]\cdot h_{siv}[\alpha - x, \beta - y]$$



2D discrete-space systems (filters)

original signal (image)

filtered signal (image)

$$f[n,m] \rightarrow \boxed{\text{System } \mathcal{S}} \rightarrow g[n,m]$$

① $g = \mathcal{S}[f]$, ② $g[n,m] = \mathcal{S}\{f[n,m]\}$

③ $f[n,m] \xrightarrow{\mathcal{S}} g[n,m]$

equivalent notation

Filters: Examples

- 2D DS moving average over a 3 × 3 window of neighborhood

$$\underbrace{g[n,m]}_{\text{new signal}} = \overbrace{\frac{1}{9}}^{\text{normalizing}} \overbrace{\sum_{k=n-1}^{n+1} \sum_{l=m-1}^{m+1}}^{3\times 3} \underbrace{f[k,l]}_{\text{original}}$$

$$= \frac{1}{9} \sum_{k=-1}^{1} \sum_{l=-1}^{1} f[n-k, m-l]$$

$$\overset{\text{filter}}{h} = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

convolution

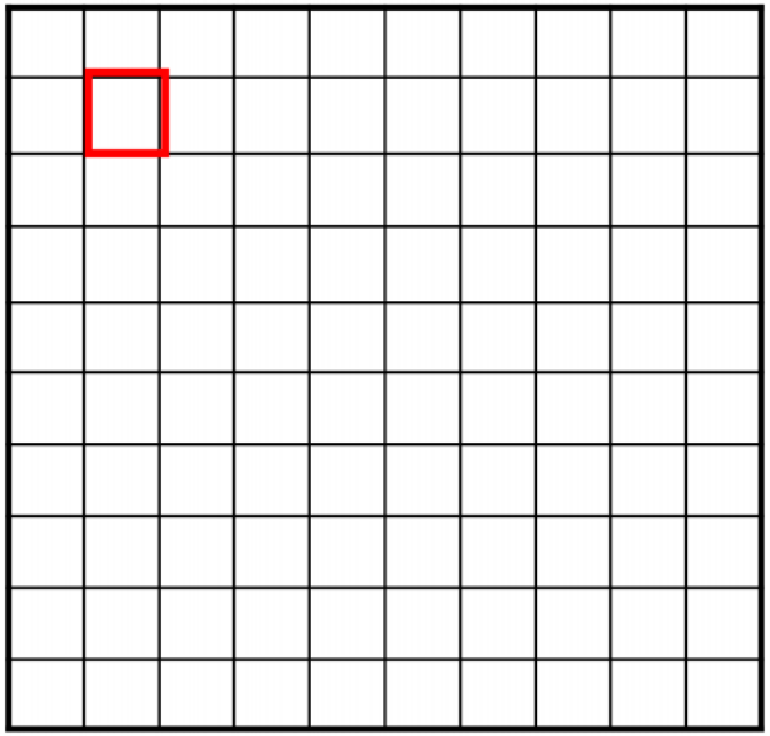
$$(f * h)[m,n] = \frac{1}{9} \sum_{k,l} f[k,l]\, h[m-k, n-l]$$

Moving average

$F[x, y]$

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

$G[x, y]$



Moving average

$F[x, y]$

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							

$$(f * g)[m,n] = \sum_{k,l} f[k,l]\, g[m-k, n-l]$$

Moving average

$F[x, y]$

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						

$$(f * g)[m,n] = \sum_{k,l} f[k,l]\, g[m-k, n-l]$$

Moving average

$F[x, y]$

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	

Moving average

In summary:

- Replaces each pixel with an average of its neighborhood.

- Achieve smoothing effect (remove sharp features)

$$g[\cdot,\cdot]$$

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

Moving average

Convolution examples

Original
Bike

Bike blurred by convolution
Impulse response „box filter“

$$\frac{1}{25}\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & [1] & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Convolution examples

Original
Bike

Bike blurred horizontally
Filter impulse response

$$\frac{1}{5}\begin{pmatrix} 1 & 1 & [1] & 1 & 1 \end{pmatrix}$$

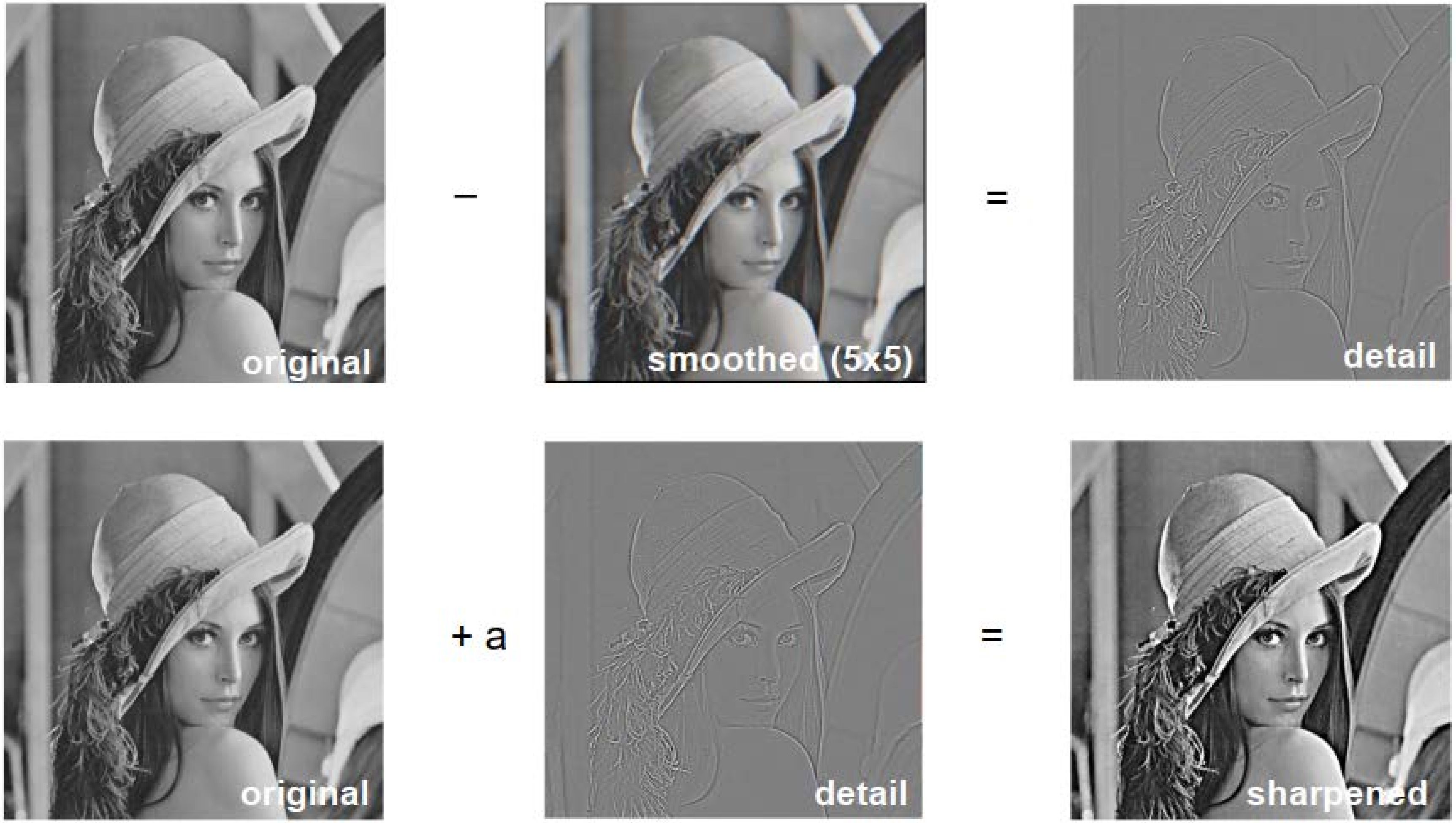
Convolution examples

Original
Bike

Bike blurred vertically
Filter impulse response

$$\frac{1}{5}\begin{pmatrix} 1 \\ 1 \\ [1] \\ 1 \\ 1 \end{pmatrix}$$

What does blurring take away?



What's More in Filtering

- Gaussian filtering by repeated box filtering
  - 1-d discrete-time Fourier transform
  - 2-d discrete-space Fourier transform
- Frequency response of 5x5 lowpass filter
  - Horizontal lowpass filter
  - Vertical lowpass filter
- Frequency response of sharpening filter
  - More aggressive sharpening

- Signals and Systems
- EE368 Lecture 8: Linear Image Processing and Filtering

Understand Convolution

- 1-d continuous:

$$(f * g)(n) = \int_{-\infty}^{\infty} f(\tau)g(n-\tau)d\tau$$

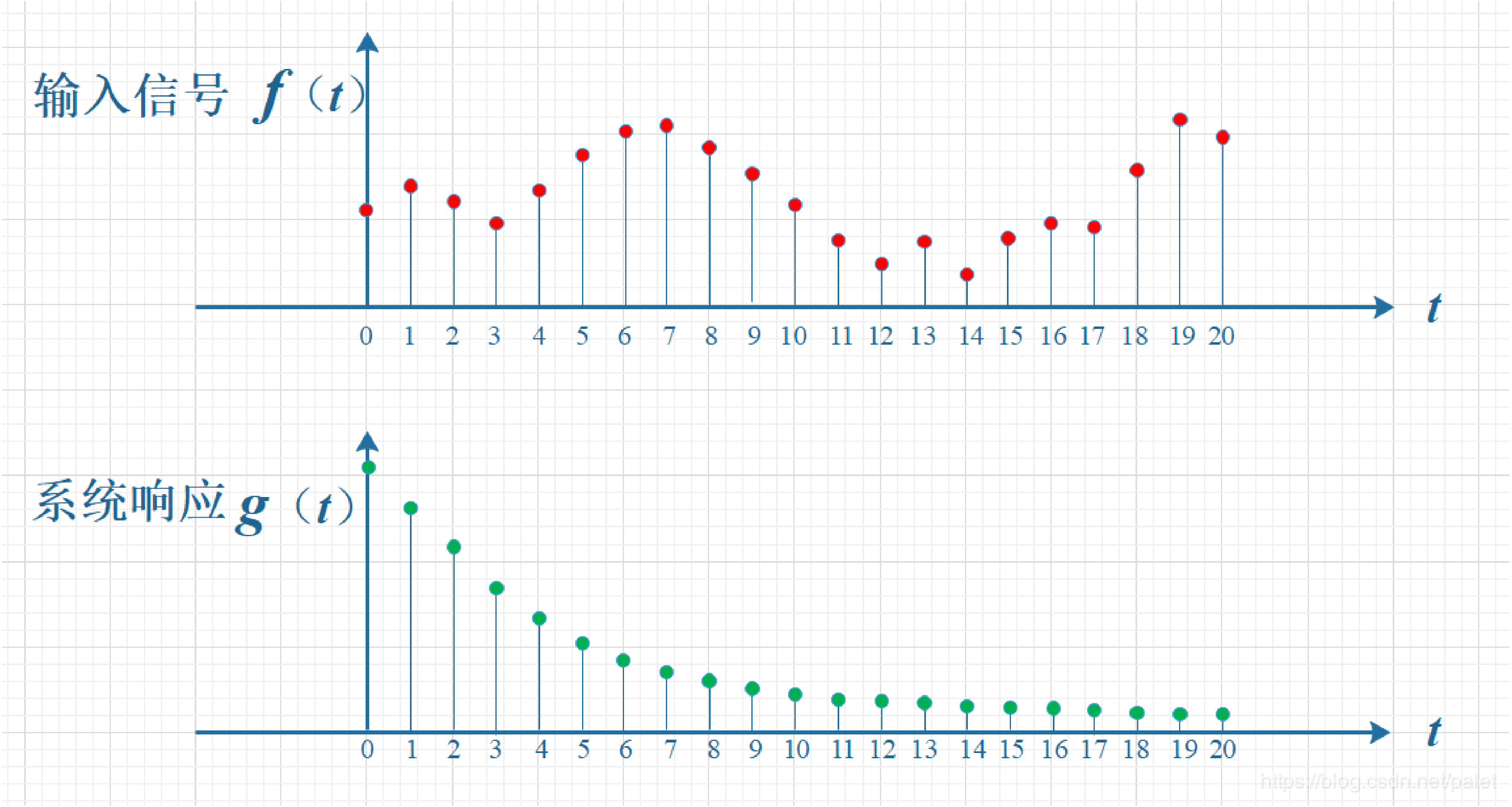
- 1-d discrete:

$$(f * g)(n) = \sum_{\tau=-\infty}^{\infty} f(\tau)g(n-\tau)$$

卷积

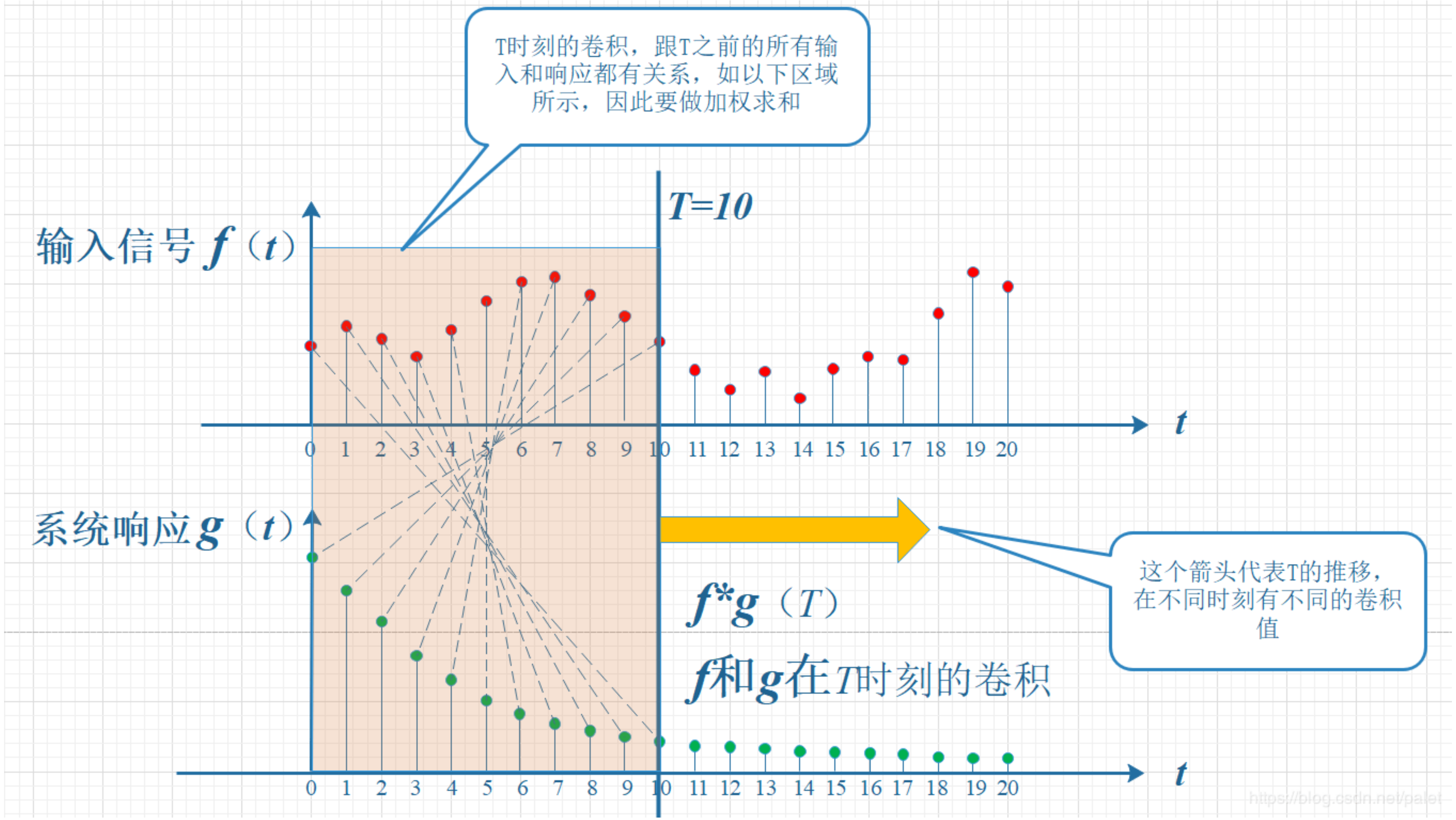
https://www.zhihu.com/question/22298352

Understand Convolution $g(\tau)$



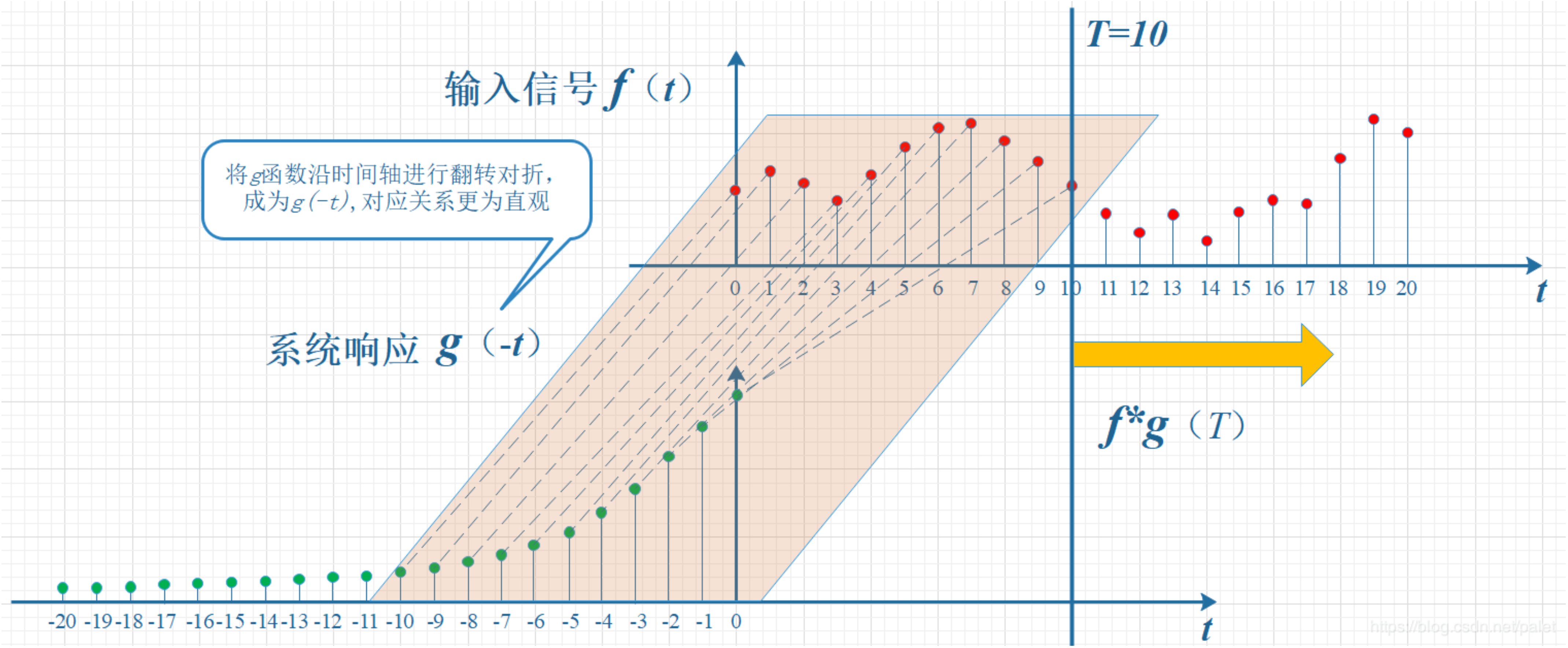
https://blog.csdn.net/palet/article/details/88862647

Understand Convolution $f * g(\tau)$



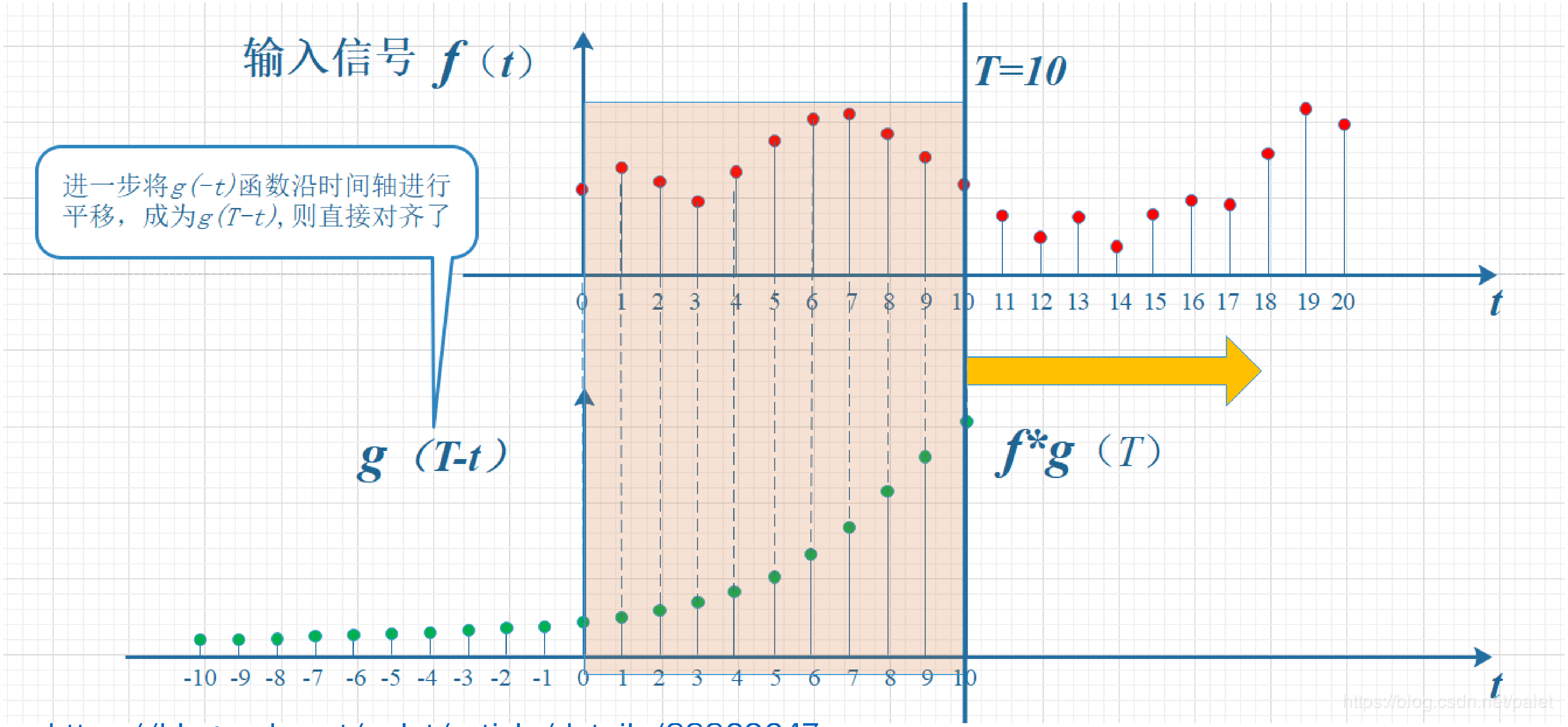
https://blog.csdn.net/palet/article/details/88862647

Understand Convolution $g(-\tau)$



https://blog.csdn.net/palet/article/details/88862647

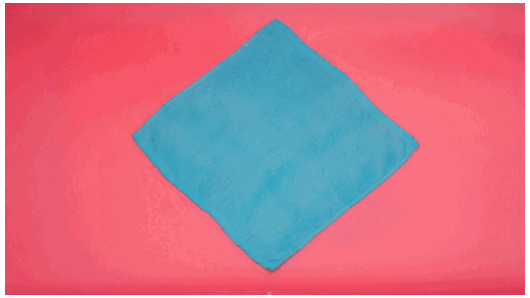
Understand Convolution $g(n-\tau)$



https://blog.csdn.net/palet/article/details/88862647

Understand Convolution Now

- Process: Flip ⟶ Slide ⟶ Superimpose (loop)
- 卷(Roll)： $g(t) \rightarrow g(-t)$. Flip
- 积(Product)： Slide and integral / sum
    - A global concept
    - Time and space mix
- Add constraint: $t + (T - t) = T$
    - Time and space explain



$$(f * g)(n) = \int_{-\infty}^{\infty} f(\tau) g(n - \tau) d\tau$$

$$n = \tau + (n - \tau)$$

$$(f * g)(n) = \sum_{\tau=-\infty}^{\infty} f(\tau) g(n - \tau)$$

https://www.matongxue.com/madocs/32.html

Another Example: Cast the dice

f	1	2	3	4	5	6

f表示第一枚骰子

$f(1)$表示投出1的概率

$f(2)$、$f(3)$、$\cdots$以此类推

g	1	2	3	4	5	6

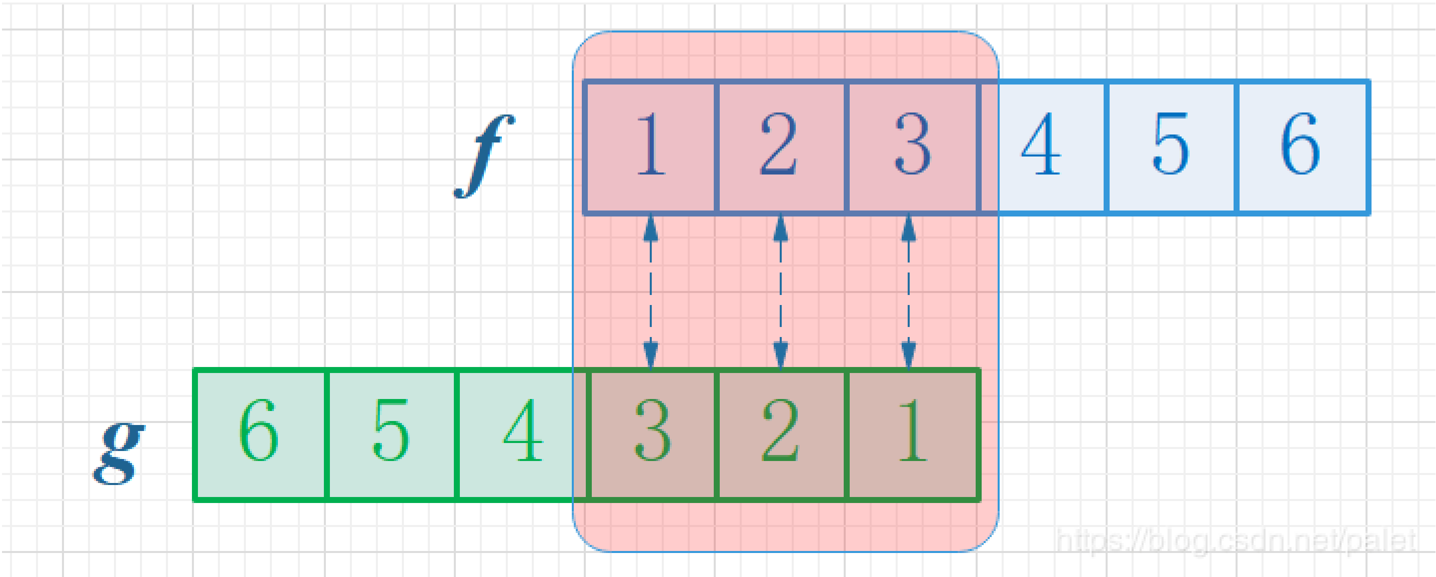
g表示第二枚骰子

两枚骰子点数加起来为 4 的概率是多少?

https://www.matongxue.com/madocs/32.html

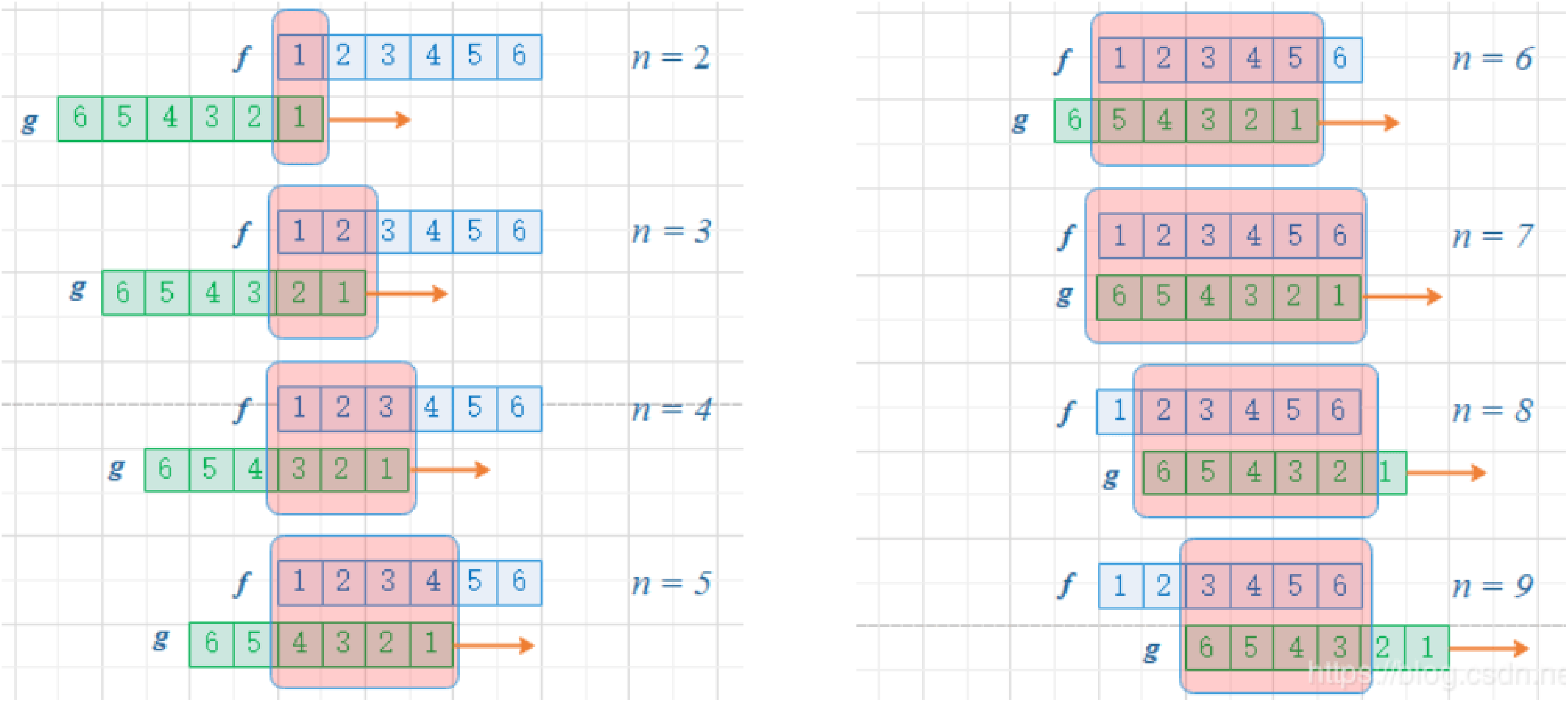
Cast the dice

- All situations: 1+3=4， 2+2=4, 3+1=4
  - $f(1)g(3) + f(2)g(2) + f(3)g(3)$
  - $(f * g)(4) = \sum_{m=1}^{3} f(4-m)g(m)$



https://www.matongxue.com/madocs/32.html

Cast the dice (Conv.)



https://www.matongxue.com/madocs/32.html

Back to Digital Image Now

$$\Longrightarrow \begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} & \cdots & a_{0,n} \\ a_{1,0} & a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,0} & a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{m,0} & a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix}$$

$$g = \begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix}$$

$$f(x,y) = a_{x,y} \quad g(x,y) = b_{x,y}$$

https://www.zhihu.com/question/22298352/answer/228543288

Digital Image Compute (Conv.)

$$f = \begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} \\ a_{1,0} & a_{1,1} & a_{1,2} \\ a_{2,0} & a_{2,1} & a_{2,2} \end{bmatrix}$$

$$\begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} & \cdots & a_{0,n} \\ a_{1,0} & \boxed{a_{1,1}} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,0} & a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{m,0} & a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix} \Longrightarrow$$

$$\begin{bmatrix} c_{0,0} & c_{0,1} & c_{0,2} & \cdots & c_{1,n} \\ c_{1,0} & \boxed{c_{1,1}} & c_{1,2} & \cdots & c_{1,n} \\ c_{2,0} & c_{2,1} & c_{2,2} & \cdots & c_{2,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ c_{m,0} & c_{m,1} & c_{m,2} & \cdots & c_{m,n} \end{bmatrix}$$

$$\Uparrow$$

$$c_{1,1} = f * g$$

https://www.zhihu.com/question/22298352/answer/228543288

NOTICE

$$f=\begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} \\ a_{1,0} & a_{1,1} & a_{1,2} \\ a_{2,0} & a_{2,1} & a_{2,2} \end{bmatrix} \qquad g=\begin{bmatrix} b_{-1,-1} & b_{-1,0} & b_{-1,1} \\ b_{0,-1} & b_{0,0} & b_{0,1} \\ b_{1,-1} & b_{1,0} & b_{1,1} \end{bmatrix}$$

a, b的下标相加都为1, 1

$$f=\begin{bmatrix} \boxed{a_{0,0}} & a_{0,1} & a_{0,2} \\ a_{1,0} & a_{1,1} & a_{1,2} \\ a_{2,0} & a_{2,1} & a_{2,2} \end{bmatrix} \qquad g=\begin{bmatrix} b_{-1,-1} & b_{-1,0} & b_{-1,1} \\ b_{0,-1} & b_{0,0} & b_{0,1} \\ b_{1,-1} & b_{1,0} & \boxed{b_{1,1}} \end{bmatrix}$$

$$(f*g)(1,1)=\sum_{k=0}^{2}\sum_{h=0}^{2} f(h,k)g(1-h,1-k)$$

$$c_{1,1}=a_{0,0}b_{1,1}$$

https://www.zhihu.com/question/22298352/answer/228543288

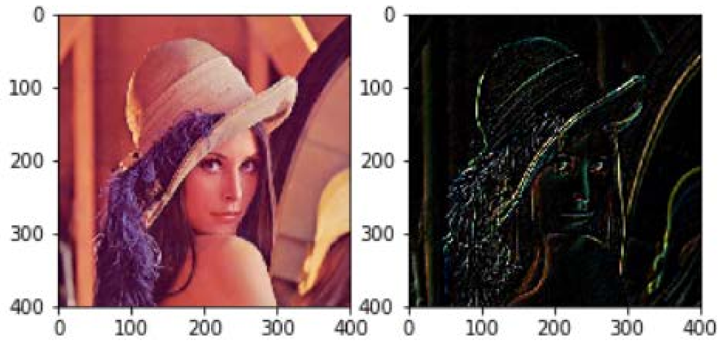
OpenCV Image Filtering

- bilateralFilter
- blur
- boxFilter
- buildPyramid
- dilate
- erode
- filter2D (notebook)
- More …

下面实现了一个简单的 3x3 卷积核，你可以试着看一下效果，尝试对其进行改进。

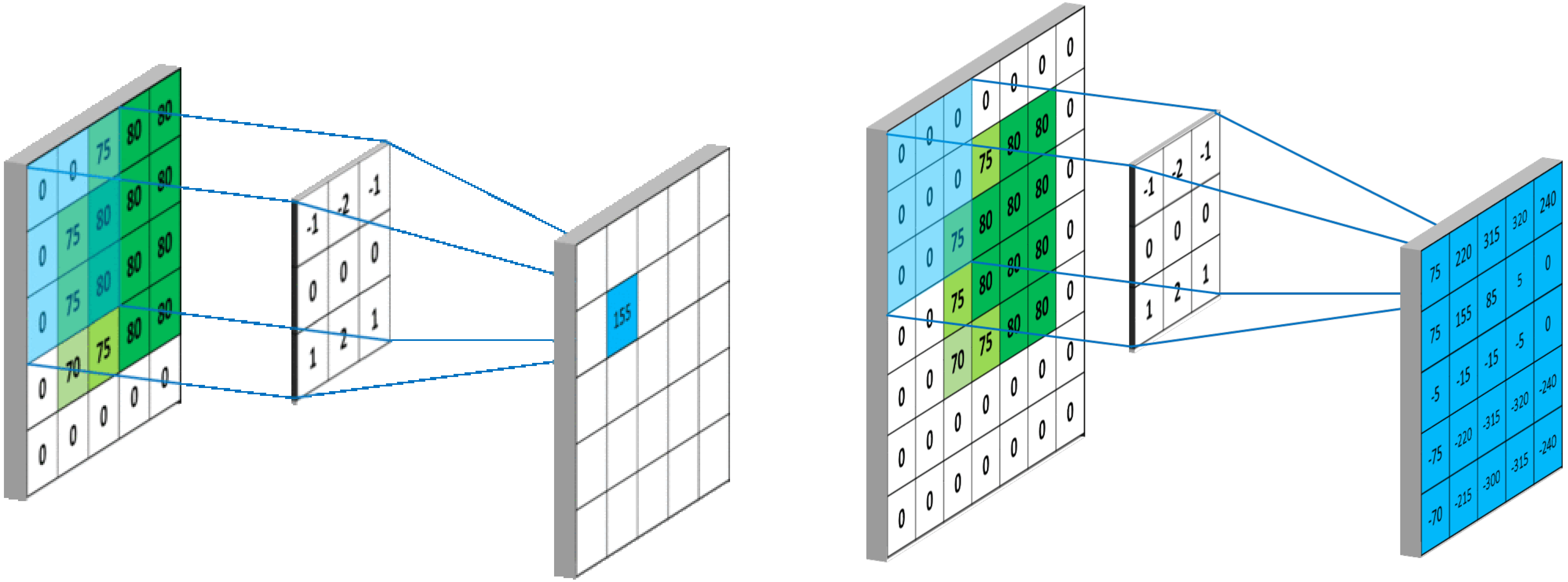
```
img = plt.imread(IMG_URL)   # 最后我要告诉你, matplotlib 本身就能读入图片, 哈哈!

plt.subplot(1,2,1)
plt.imshow(img)

fil = np.array([[ -1,-1, 0],                        #这个是设置的滤波, 也就是卷积核
                [ -1, 0, 1],
                [  0, 1, 1]])

res = cv2.filter2D(img,-1,fil)                       #使用opencv的卷积函数

plt.subplot(1,2,2)
plt.imshow(res)                                      #显示卷积后的图片
```

<matplotlib.image.AxesImage at 0x23807a01208>



https://docs.opencv.org/3.0-beta/modules/imgproc/doc/filtering.html

Conv layer in Deep Neural Network



https://mlnotebook.github.io/post/CNN1/

Convolution vs. Correlation

- A convolution is an integral that expresses the amount of overlap of one function as it is shifted over another function.
  - convolution is a filtering operation

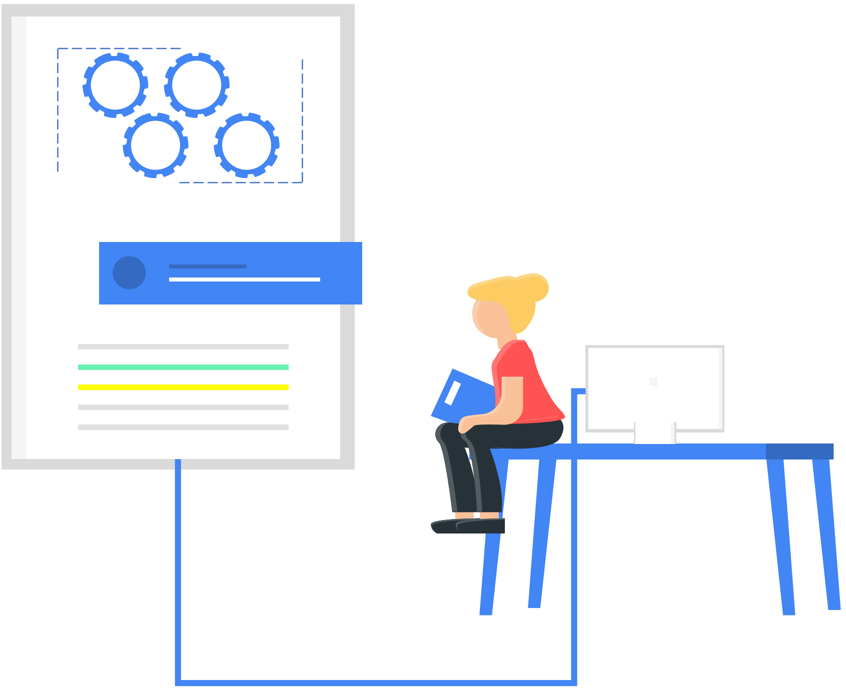
- Correlation compares the *similarity* of *two* sets of *data*. Correlation computes a measure of similarity of two input signals as they are shifted by one another. The correlation result reaches a maximum at the time when the two signals match best .
  - correlation is a measure of relatedness of two signals

- We'll talk more about correlation in next section.

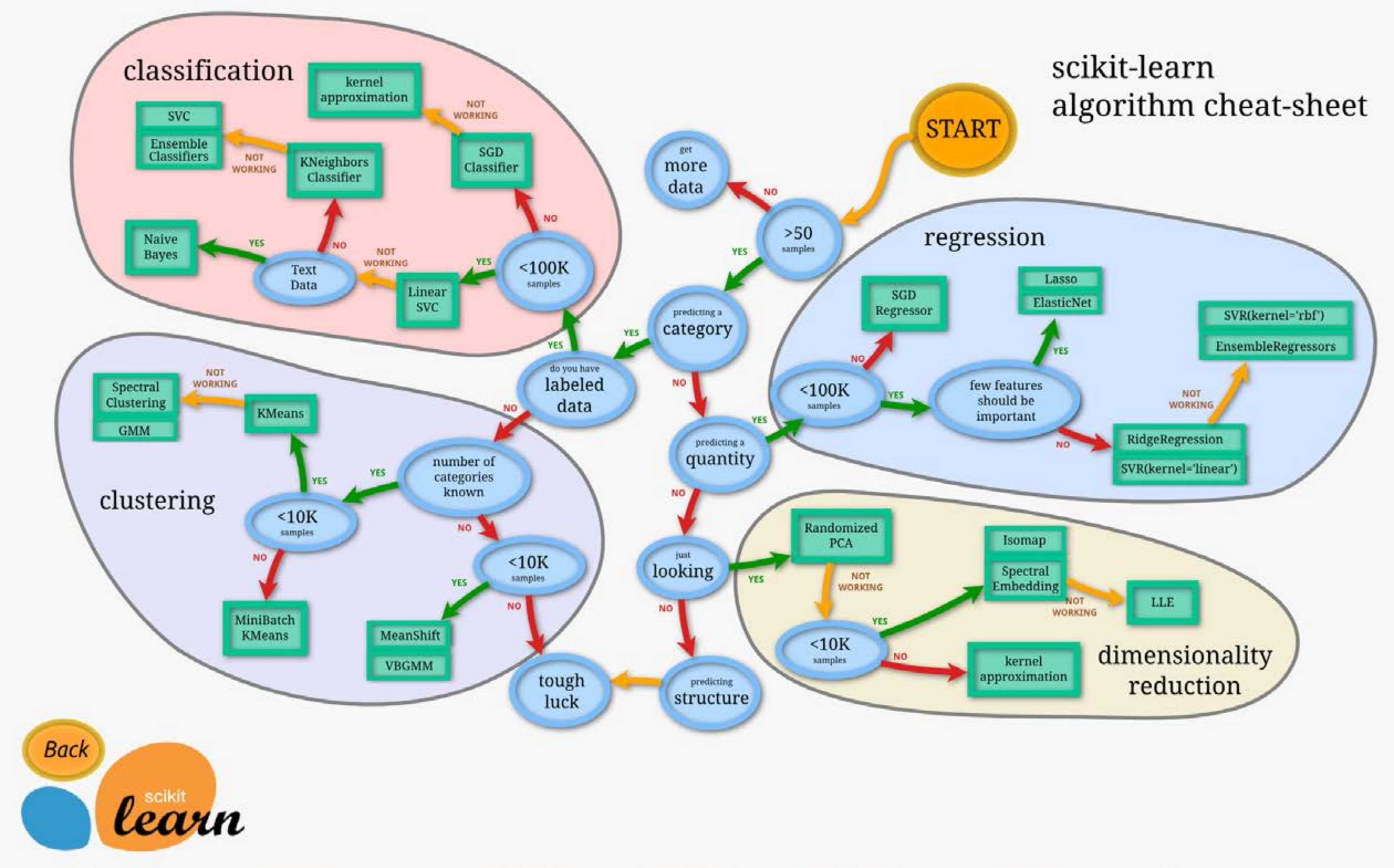
Break

5 mins

Contents



Choosing the right estimator



https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

Practice the Algorithm

- Machine Learning in Python – scikit-learn

Classification

Identifying to which category an object belongs to.

Applications: Spam detection, Image recognition.
Algorithms: SVM, nearest neighbors, random forest, …

— Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: SVR, ridge regression, Lasso, …

— Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, …

— Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency
Algorithms: PCA, feature selection, non-negative matrix factorization.

— Examples

Model selection

Comparing, validating and choosing parameters and models.

Goal: Improved accuracy via parameter tuning
Modules: grid search, cross validation, metrics.

— Examples

Preprocessing

Feature extraction and normalization.

Application: Transforming input data such as text for use with machine learning algorithms.
Modules: preprocessing, feature extraction.

— Examples

Example: Decision Trees

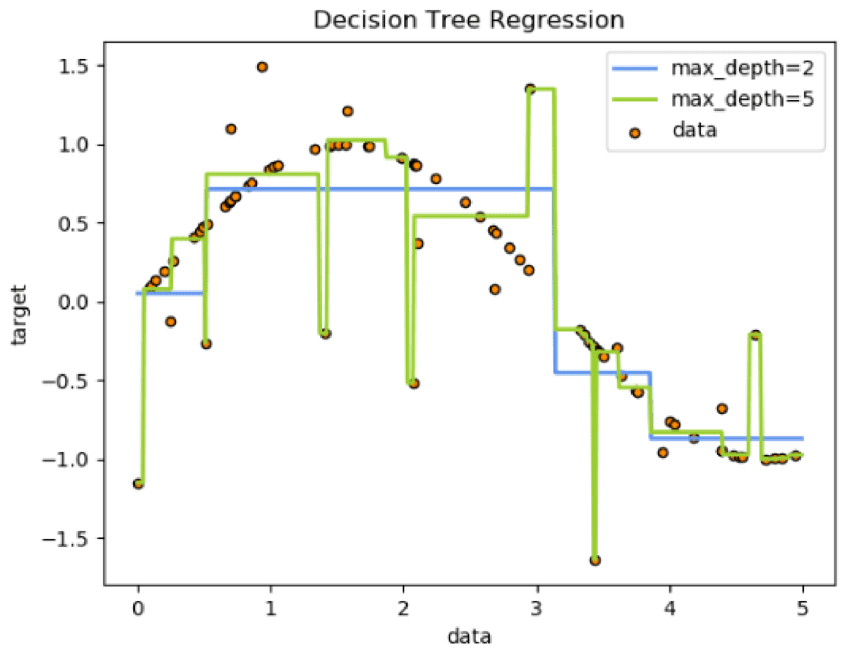
scikit-learn v0.21.2
Other versions

Please **cite us** if you use the software.

1.10. Decision Trees

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

For instance, in the example below, decision trees learn from data to approximate a sine curve with a set of if-then-else decision rules. The deeper the tree, the more complex the decision rules and the fitter the model.



Example: Decision Trees

Using the Iris dataset, we can construct a tree as follows:

```
>>> from sklearn.datasets import load_iris
>>> from sklearn import tree
>>> iris = load_iris()
>>> clf = tree.DecisionTreeClassifier()
>>> clf = clf.fit(iris.data, iris.target)
```

Once trained, you can plot the tree with the plot_tree function:

```
>>> tree.plot_tree(clf.fit(iris.data, iris.target))
```

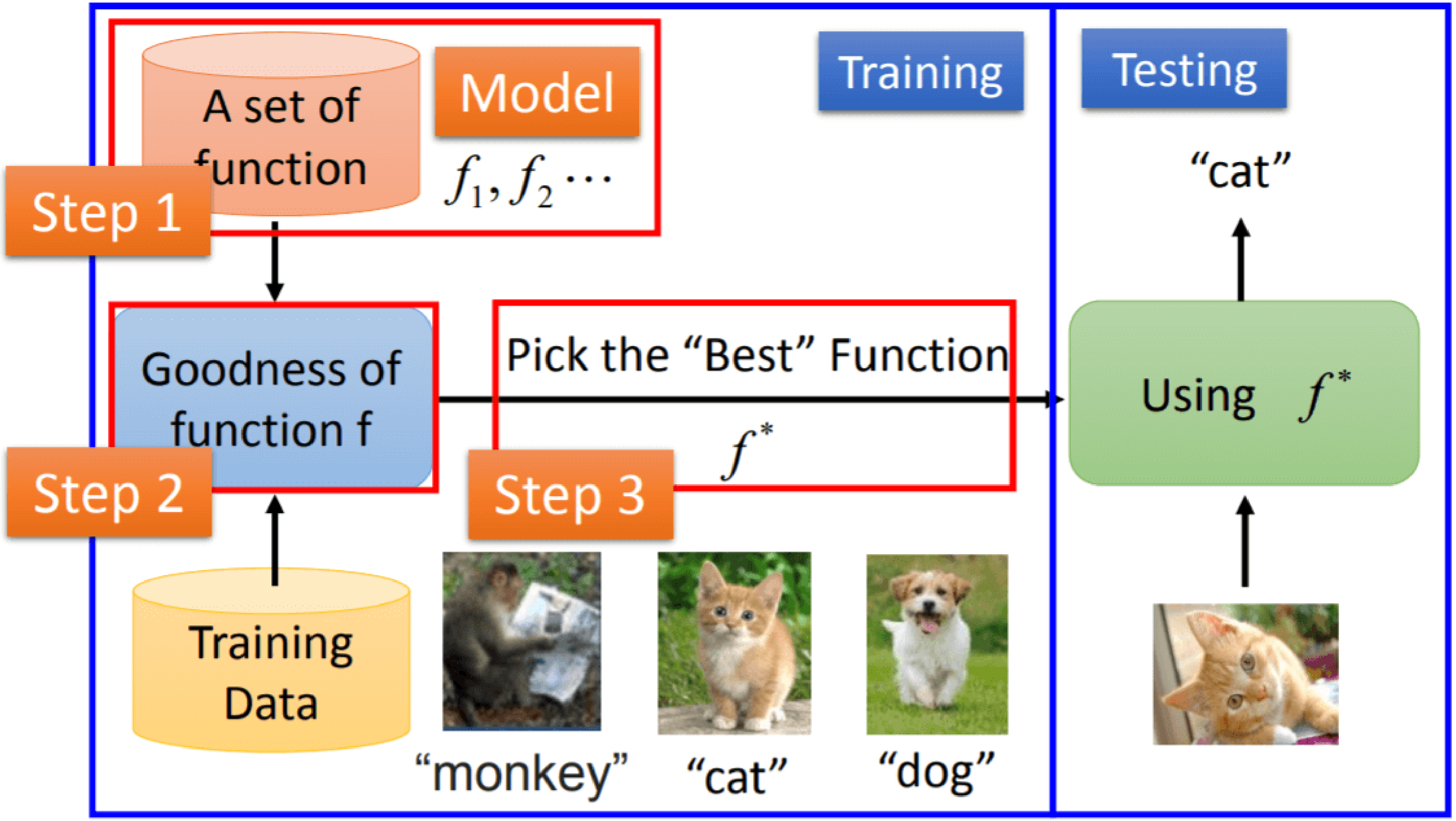
Examples:

- Multi-output Decision Tree Regression
- Face completion with a multi-output estimators

References:

- M. Dumont et al, Fast multi-class image annotation with random subwindows and multiple output randomized trees, International Conference on Computer Vision Theory and Applications 2009

Machine Learning Basic Concepts



http://speech.ee.ntu.edu.tw/~tlkagk/courses_ML17_2.html

Advice for Linear Models Preview

- 助教的博客(中文，对新手友好):
  - 线性代数回顾 – 矩阵运算 / 矩阵微积分
  - 概率论回顾 – 尤其是理解概率分布(后面可能会讨论)
  - 线性回归与梯度下降 – 线性回归，结合西瓜书推导
  - 梯度下降细节与技巧 – 深入了解梯度下降
  - 偏差与方差——误差从何而来？ - 对于理解损失有帮助
  - 线性分类与逻辑回归 – 简单的逻辑回归介绍
  - 机器学习思想比较 - 提到了如何看 PRML，理解贝叶斯视角
  - 数学符号的严谨性说明(还没有写)
  - 高屋建瓴之线性回归 – 高观点视角的线性回归
  - 机器学习模型发展 – 学习如何学习！

Blogs for Learn SVM

- Pluskid: 支持向量机系列(需要了解一些凸优化知识)
  - Maximum Margin Classifier — 支持向量机简介。
  - Support Vector —目标函数的 dual 优化推导，并得出“支持向量”的概念。
  - Kernel — 介绍核方法，并由此将支持向量机推广到非线性的情况。
  - Outliers — 介绍支持向量机使用松弛变量处理 outliers 方法。
  - Numerical Optimization — 简要介绍求解求解 SVM 的数值优化算法。
  - Duality — 关于 dual 问题推导的一些补充理论。
  - Kernel II — 核方法的一些理论补充，关于 Reproducing Kernel Hilbert Space 和 Representer Theorem 的简介。
  - Regression — 关于如何使用 SVM 来做 Regression 的简介(没更新...)
- 不要看拼凑而成的博客，行文缺乏主要思想

New TRY in next Sections

- A lot of Input: … Read / Watch / Coding
- Try output:
  - Write down Notes & Coding
  - Organize an article (再谈博客设计与写作)
- Give a public Presentation(PRE)
  - Content
  - Logic
  - Interaction
- In the future: piazza gradescope

Self-learning Tasks

课外	-	**跳坑，爬坑，填坑，避坑** 如何利用前人经验快速解决 Bug? 如何整理笔记，撰写技术报告? 如何将个人笔记进一步整理为博客? 如何制作学术汇报型幻灯片? 如何在幻灯片中使用 $\LaTeX$ 公式? 创作的时候如何尊重知识产权? [PPT 带母版模板] [LaTeX 成品举例] [PPT 成品举例 - AI for Everyone]	**官方网站:** StackOverflow CSDN 博客园 Markdown(Wiki) Typora LaTeX(Wiki) Power Point Google Slides Overleaf **参考资料:** 中文技术文档的写作规范 再谈博客设计与写作 [MD 教程] 什么是幻灯片母版? 使用键盘快捷方式创建 PPT IguanaTex - LaTeX Add-In for PPT. 怎样做好学术 PPT?	讨论效果以内容完整，思路清晰为主，不必过于注重形式。文章写作时，建议初期使用 Markdown 语法快速梳理内容，必须的时候再进阶使用 LaTeX 语法排版，英文论文发表需要熟练使用后者。

Let’s show them...

Have a nice weekend~

"Be curious. Read widely. Try new things.
What people call intelligence just boils down
to **curiosity**." -- Aaron Swartz