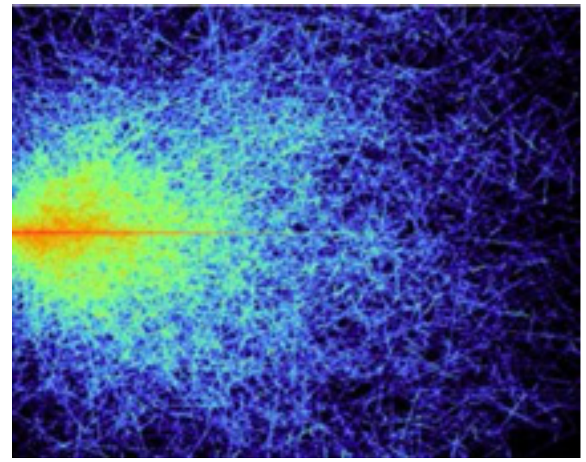
GDR BioComp, Bordeaux, June 4 - 6, 2018



From **computational imaging** to **optical computing**

Laurent Daudet, CTO at LightOn

LightOn

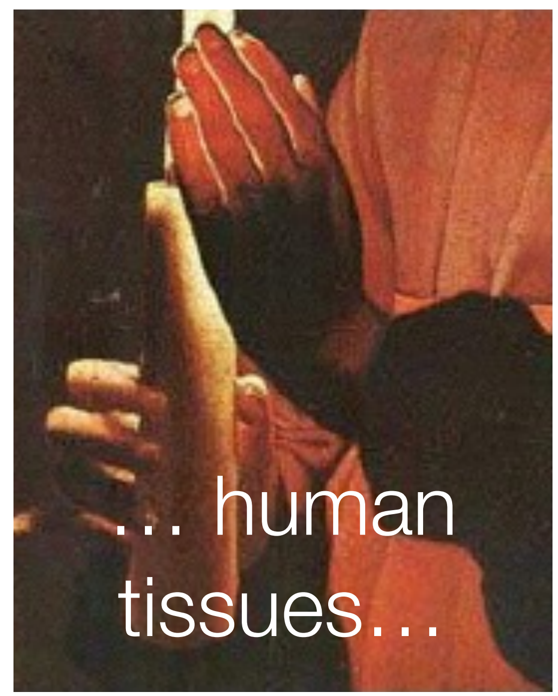
www.LightOn.io

laurent@lighton.io

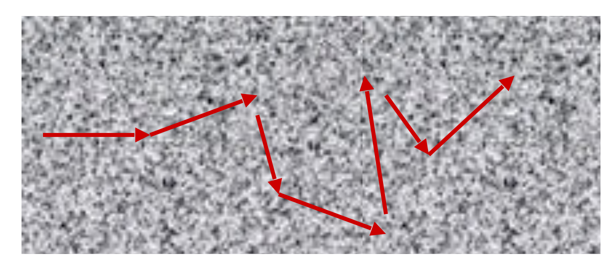
@Laurent_Daudet
@LightOnIO

Light scattering by diffusive materials

Part of our everyday experience:



Origin: light is scattered by inhomogeneities



Spoiler :

Multiple light scattering through diffusive materials is an extremely complicated process that can be described on a macroscopic level and under coherent light by extremely simple equations

… with more than 10^{12} parameters

It is then possible to leverage this to :

- perform new optics through « computational imaging »
- design new « optical computing » paradigms

A combination of expertise from:

- optics
- signal processing
- optimization / machine learning

Sylvain Gigan

LKB (UPMC / ENS)

Florent Krzakala

LPS (UPMC / ENS)

Igor Carron

Nuit Blanche / LightOn

And many others from their research teams and at LightOn

Outline

How to

- … get Superman vision
- … learn from the blur
- … make pythons crawl faster

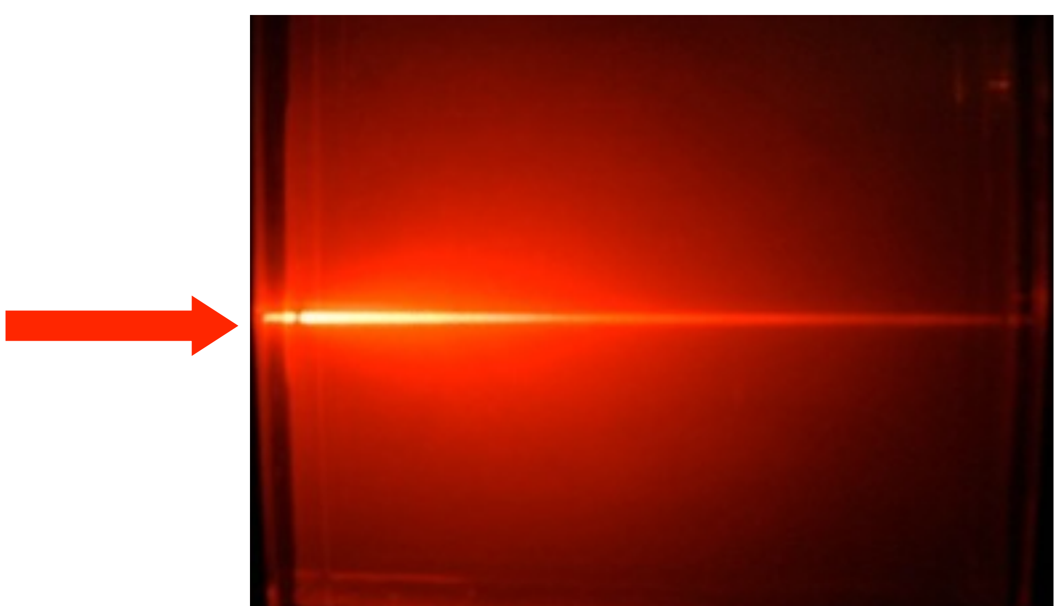
Outline

How to

- … get Superman vision
- … learn from the blur
- … make pythons crawl faster

Imaging in scattering media

Conventionally : information from only unscattered (*'ballistic'*) light



Beer-Lambert Law: Exponential decay of the ballistic light

→ No imaging beyond a few hundred microns in living tissues

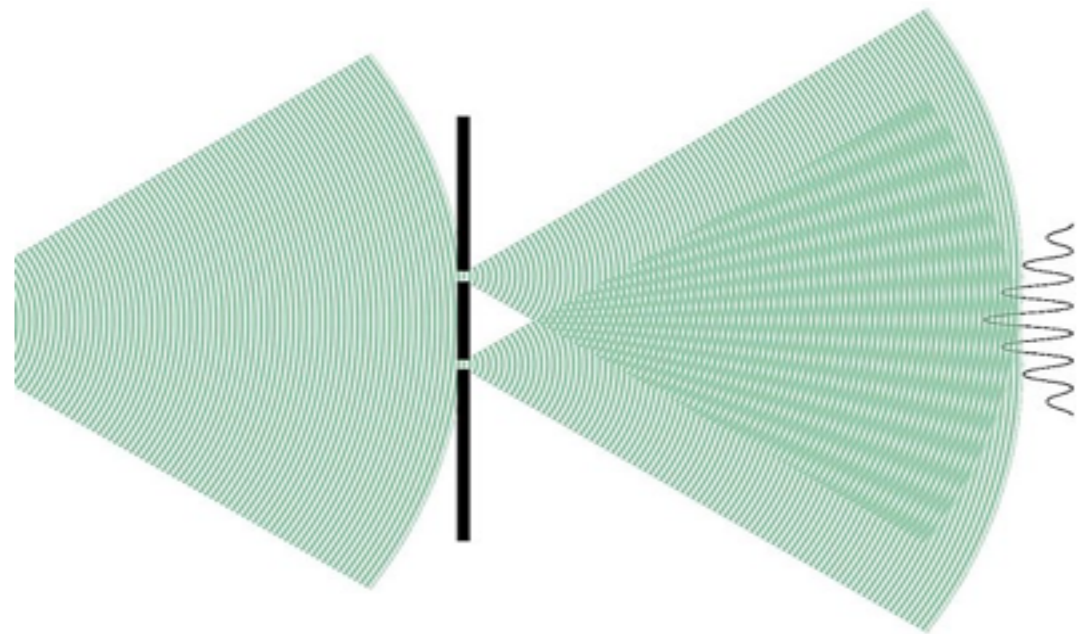
CAN WE GO DEEPER?

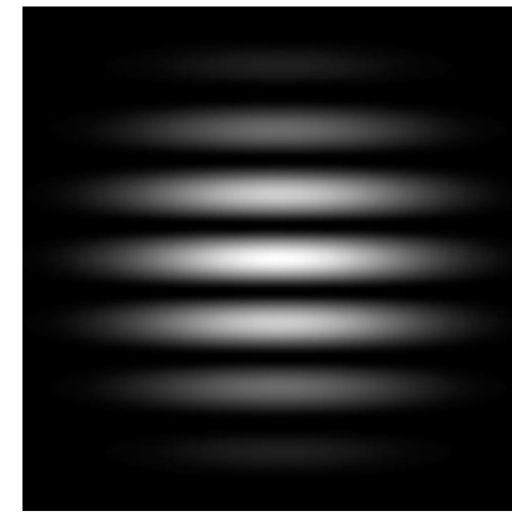
Scattering : a coherent process

Young's slit experiment:

two wave interference

Fringes

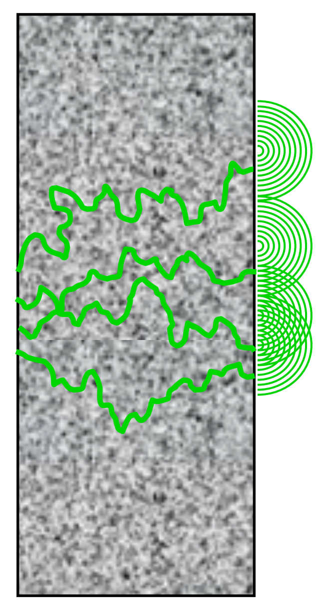


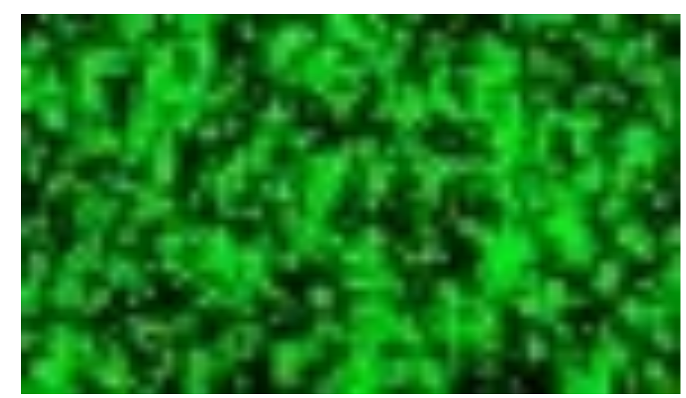


Volume scattering:

Coherent light
(laser)

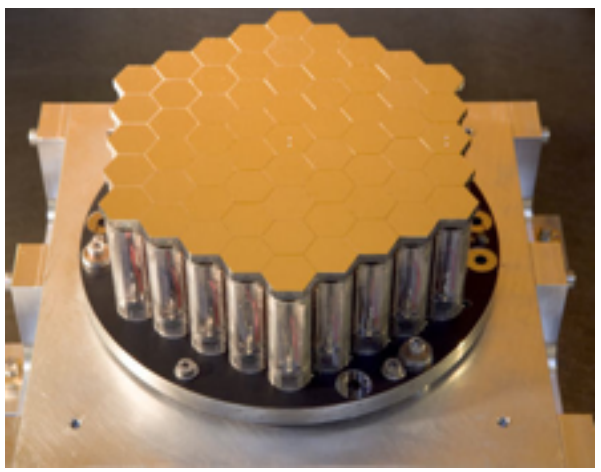
thin layer (about 300 μm)
of white paint
(particle size ≤ 1 μm)





Speckle results from multiple interference
between a multiplicity of random paths

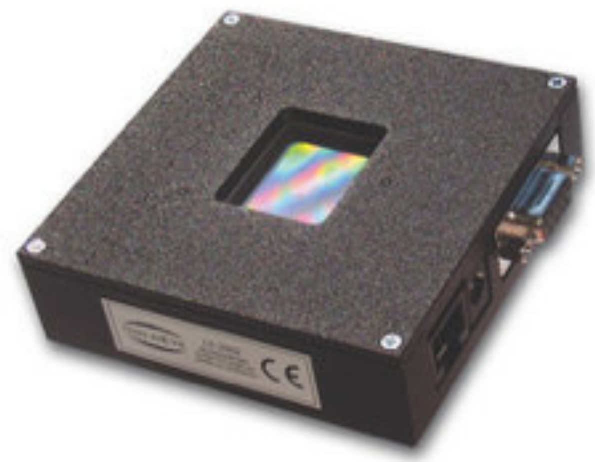
Wavefront shaping: a tool to study scattering



Deformable mirrors

10-100 actuators
moving: 10-20 microns
Speed > kHz

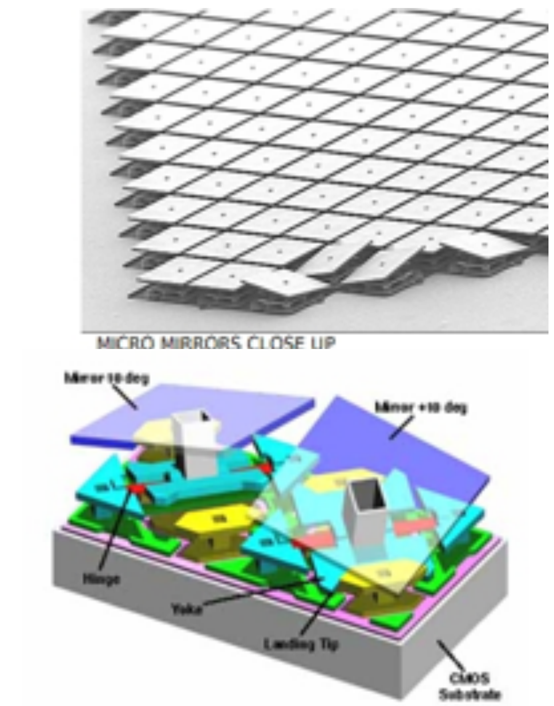
Adaptive optics



Spatial Light Modulators
based on Liquid crystals

>1 million pixels
Phase modulation at: 50Hz

Display



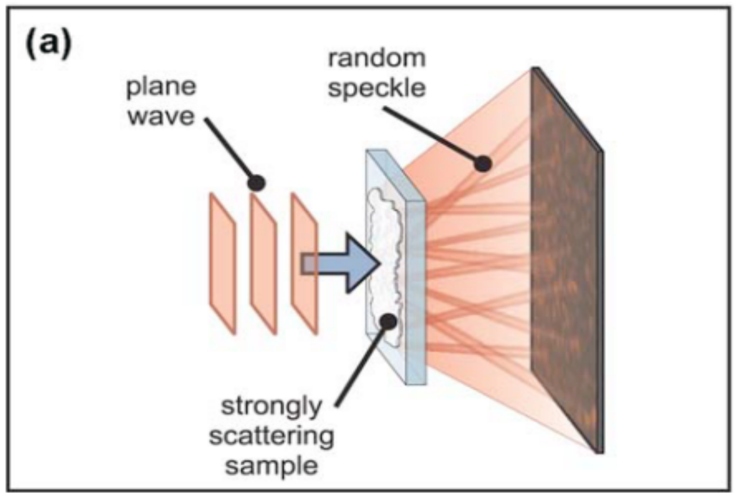
Spatial light modulators based on
MEMS technology
ex: Texas DLP/DMD

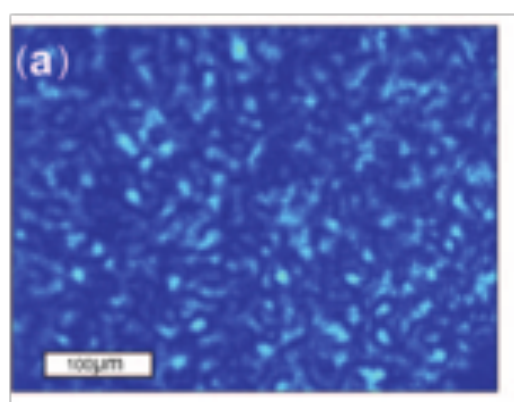
>1 million pixels
binary ON/OFF at 20kHz

Display

LightOn

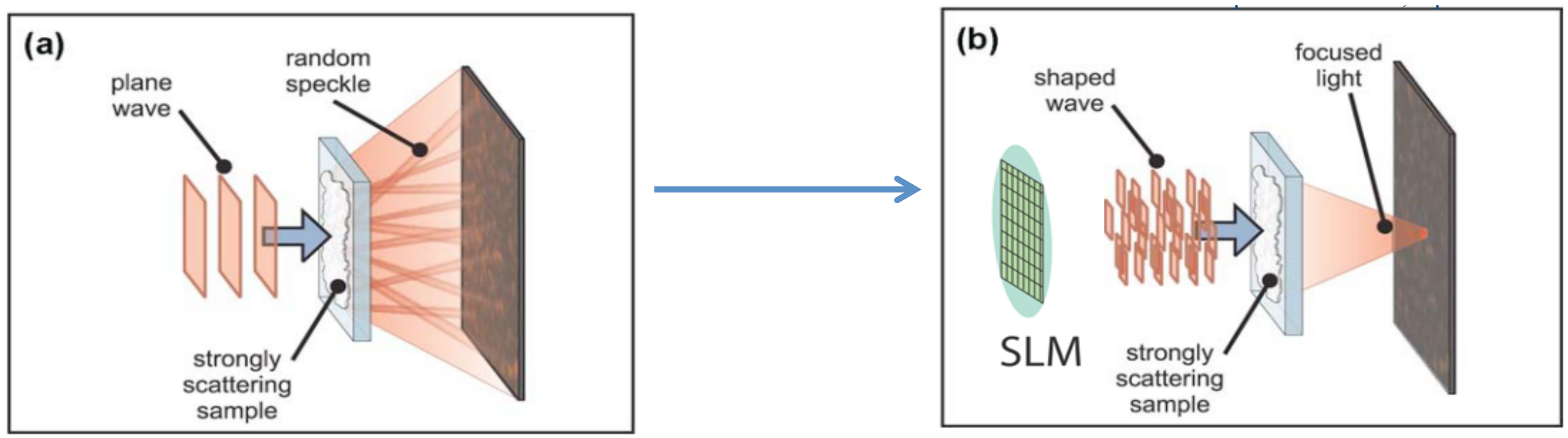
Focusing by Optimization

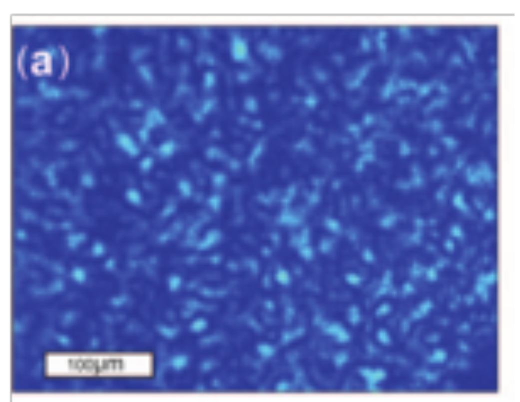


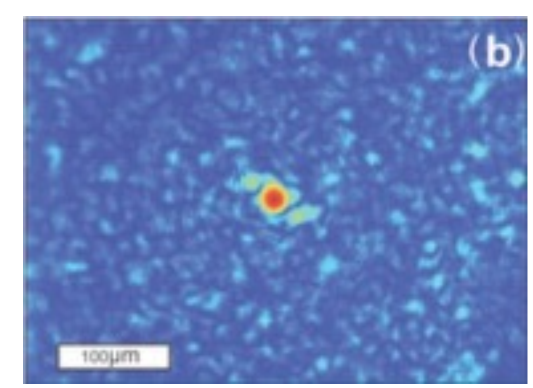


Focusing by Optimization

IM Vellekoop and AP Mosk, Optics Letters, 32(16) 2007



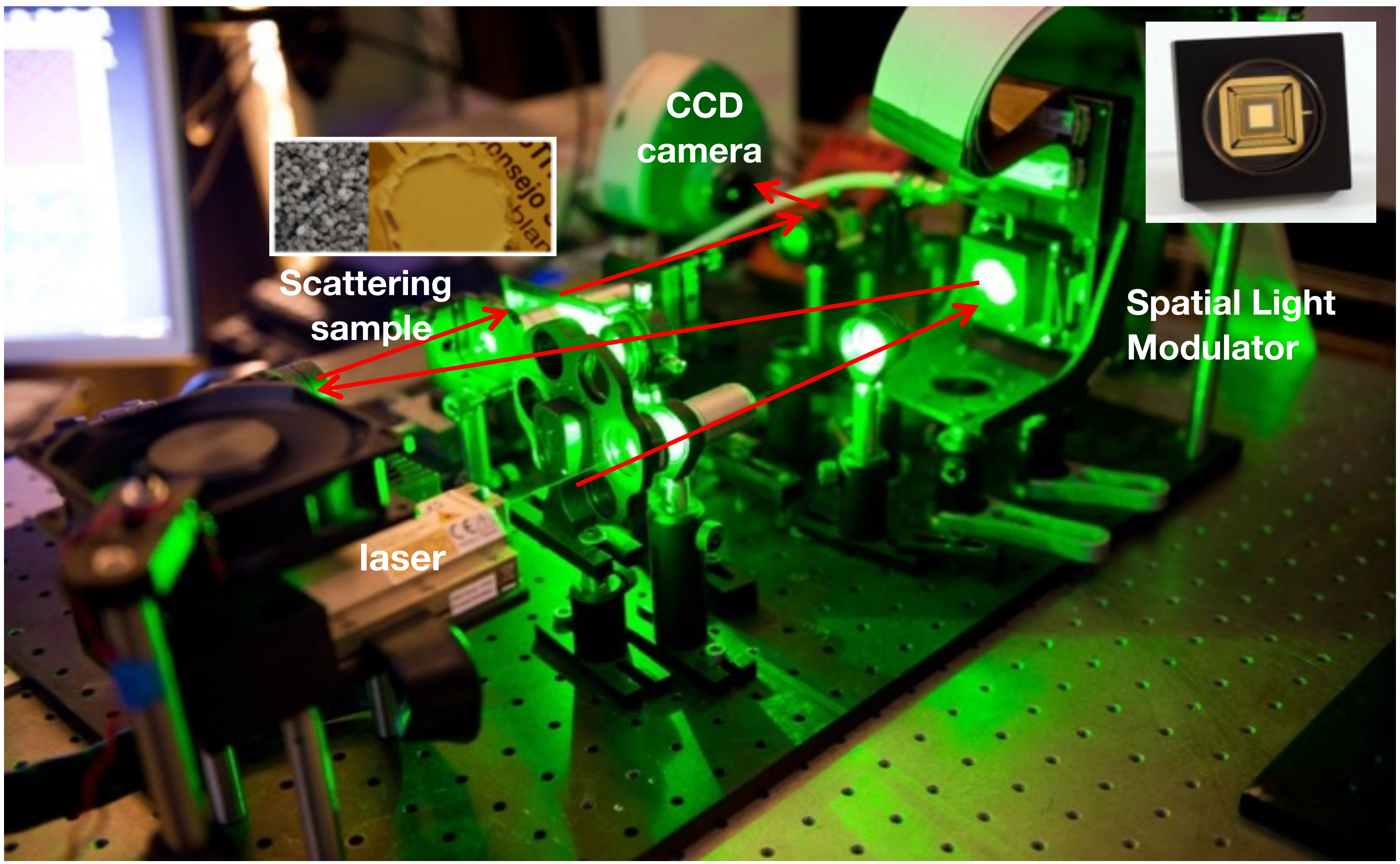




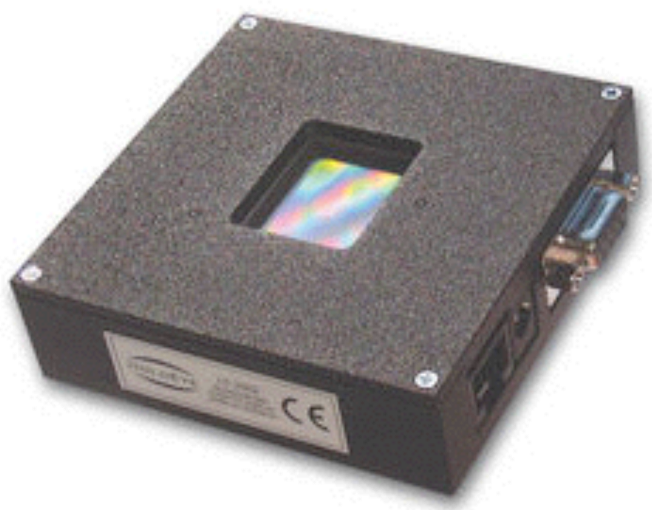
It is possible to shape the incoming wavefront to obtain a constructive interference on a single speckle grain « turn paint into a lens »

Focusing by Optimization

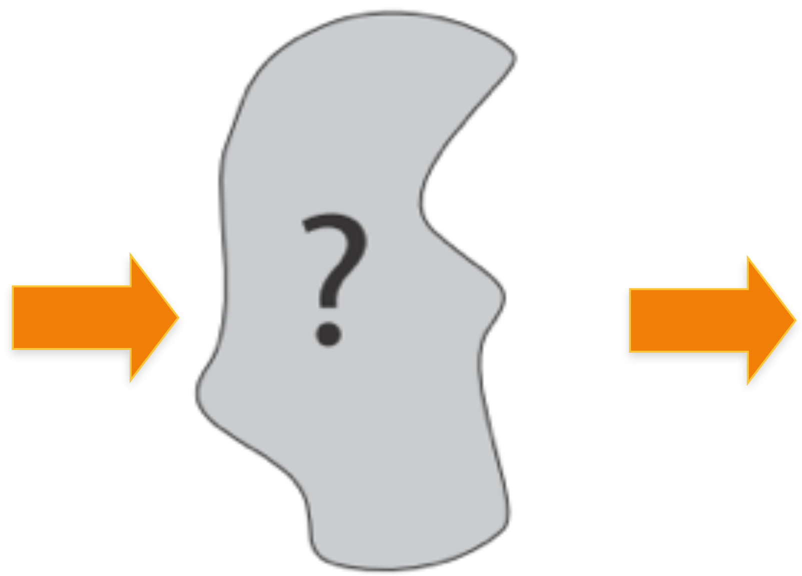
in the lab of Sylvain Gigan - ENS / LKB



General approach: the transmission matrix



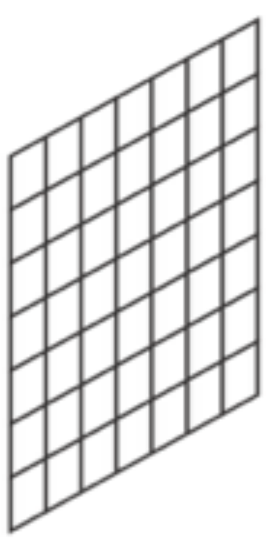
SLM: array of pixels



Linear system

CCD camera: arrays of pixels

=



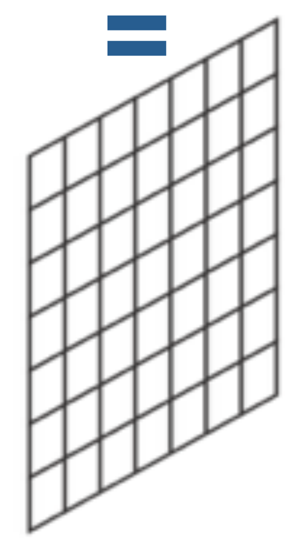
N complex-valued amplitudes

=

$$H = \begin{pmatrix} h_{1,1} & h_{1,2} & \dots & h_{1,N} \\ h_{2,1} & h_{2,2} & \dots & h_{2,N} \\ \vdots & & \ddots & \vdots \\ h_{M,1} & h_{M,2} & \dots & h_{M,N} \end{pmatrix}$$

MxN complex-valued matrix

=



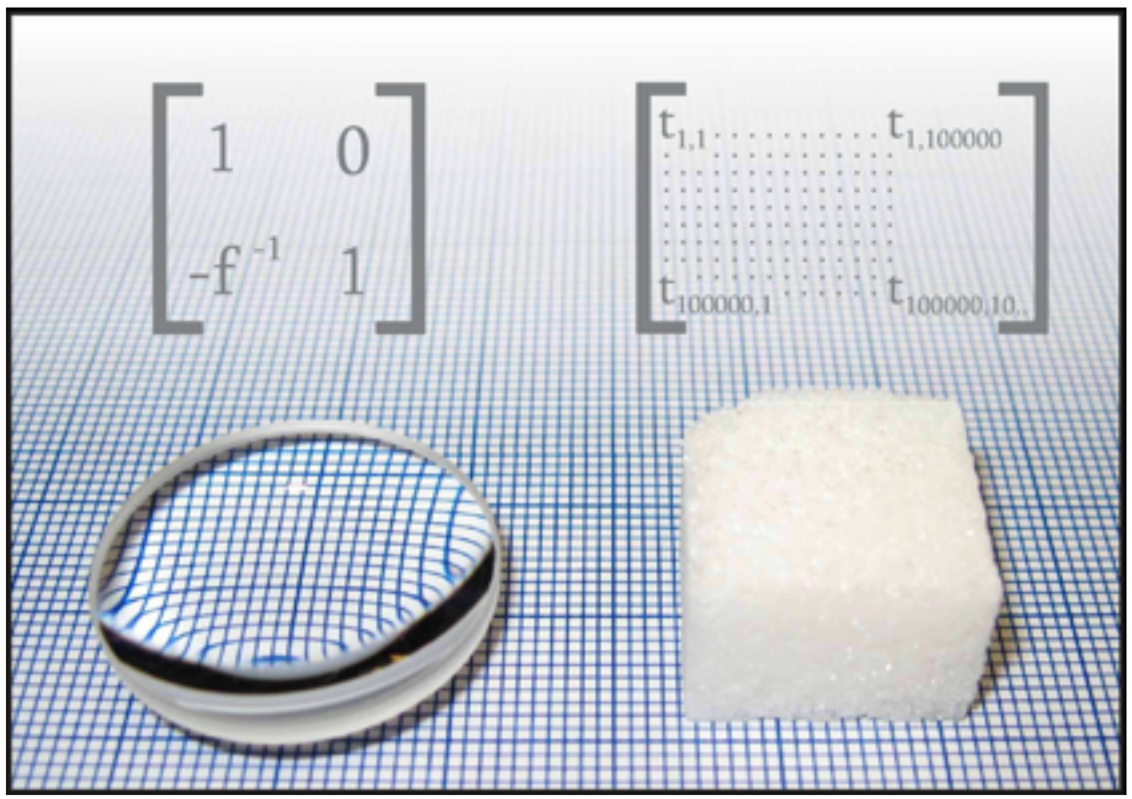
M modulus of complex-valued coefficients

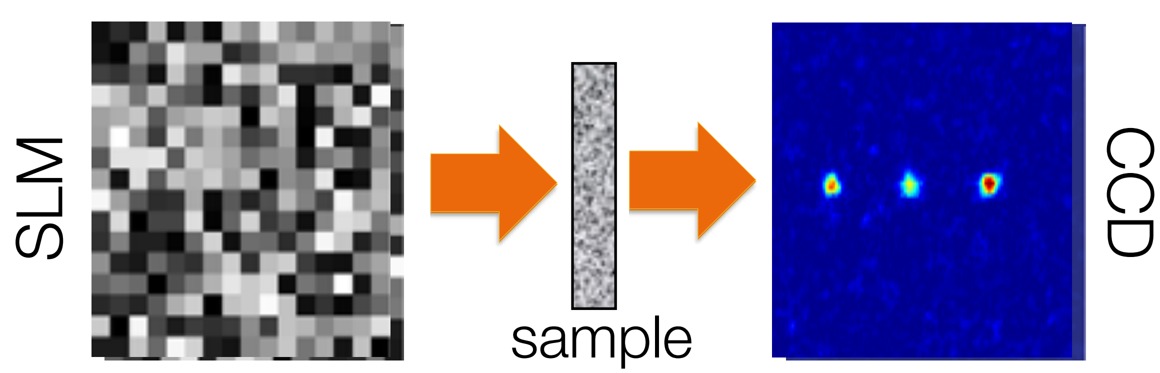
$$\left| E_m^{out} \right| = \left| \sum_n h_{mn} E_n^{in} \right|$$

Popoff et al. *Nat. Commun.* 1:81
doi: 10.1038/ncomms1078 (2010)

General approach: the transmission matrix

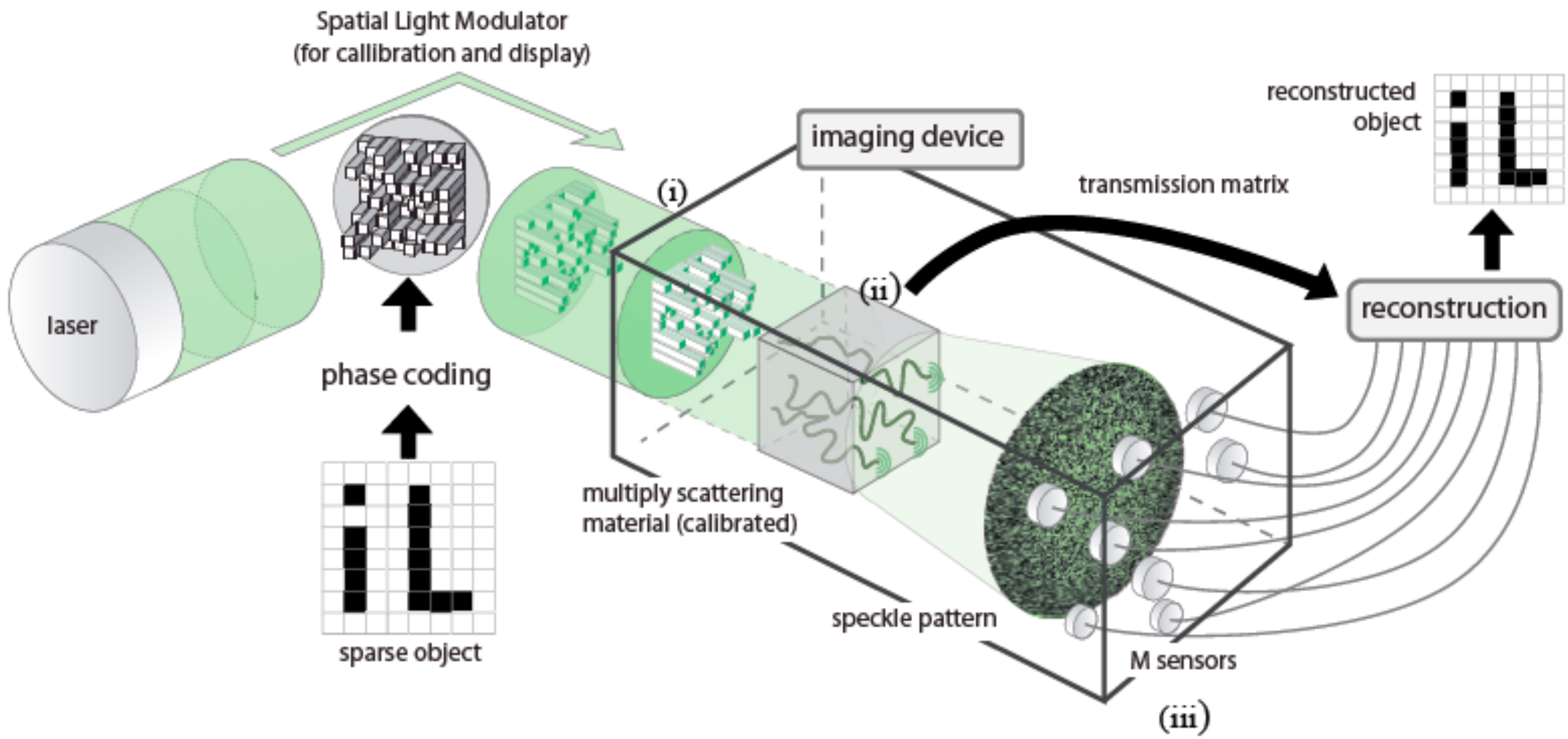
knowing the transmission matrix turns the scattering material into a « lens » with a very high number of degrees of freedom





Popoff et al. Phys. Rev. Lett. 104,100601 (2010)

Compressive imaging with scattering media



Proof of concept for **compressive imaging** with simple hardware

Take-home message Part 1

- It is possible to « see » through a strongly scattering material
- Volume scattering preserves the information content
- It « optimally » mixes information, evenly spread on output pixels: all samples are created equal !

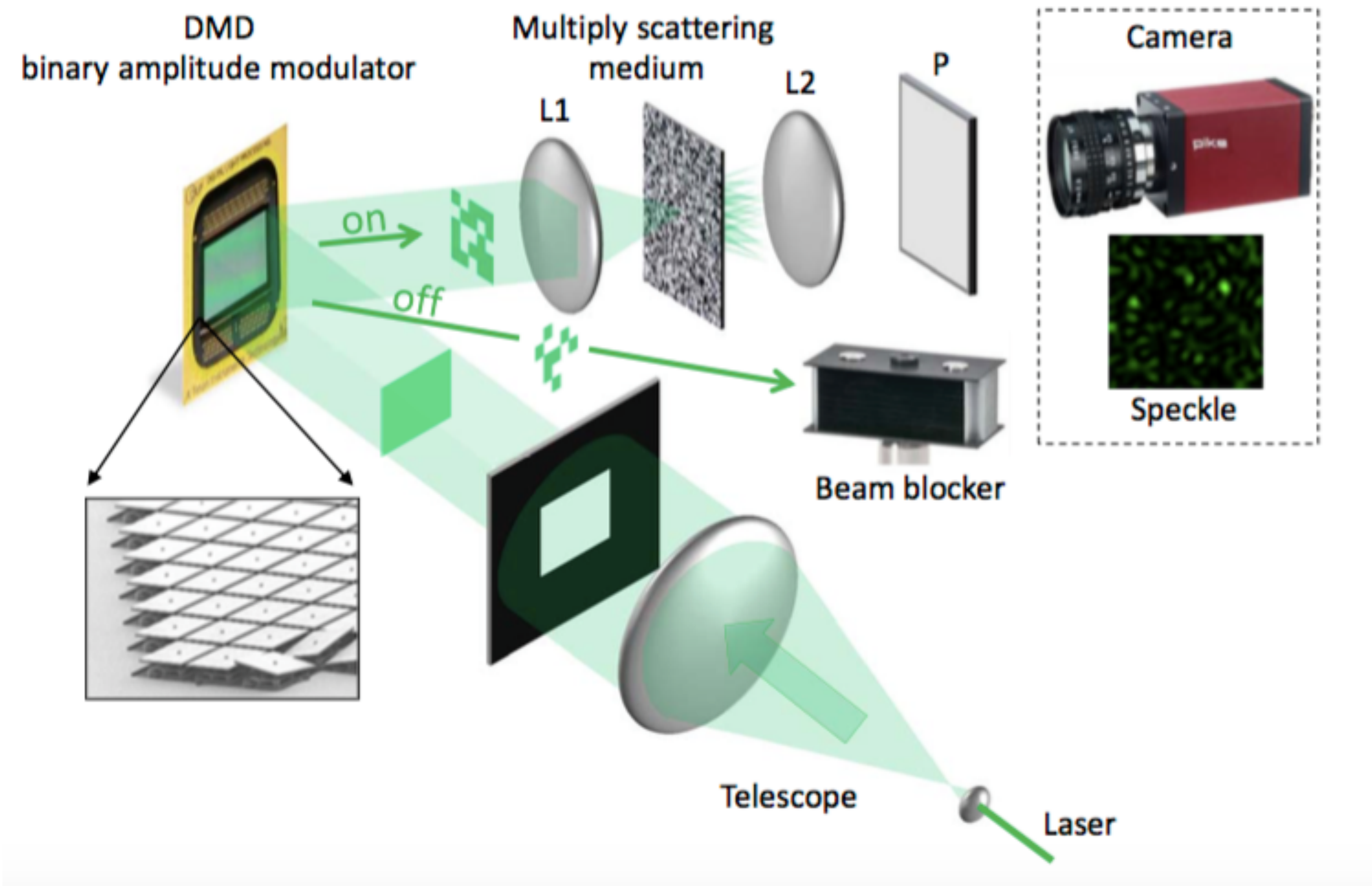
« Ask not what computing can do for optics – ask what optics can do for computing »

Outline

How to

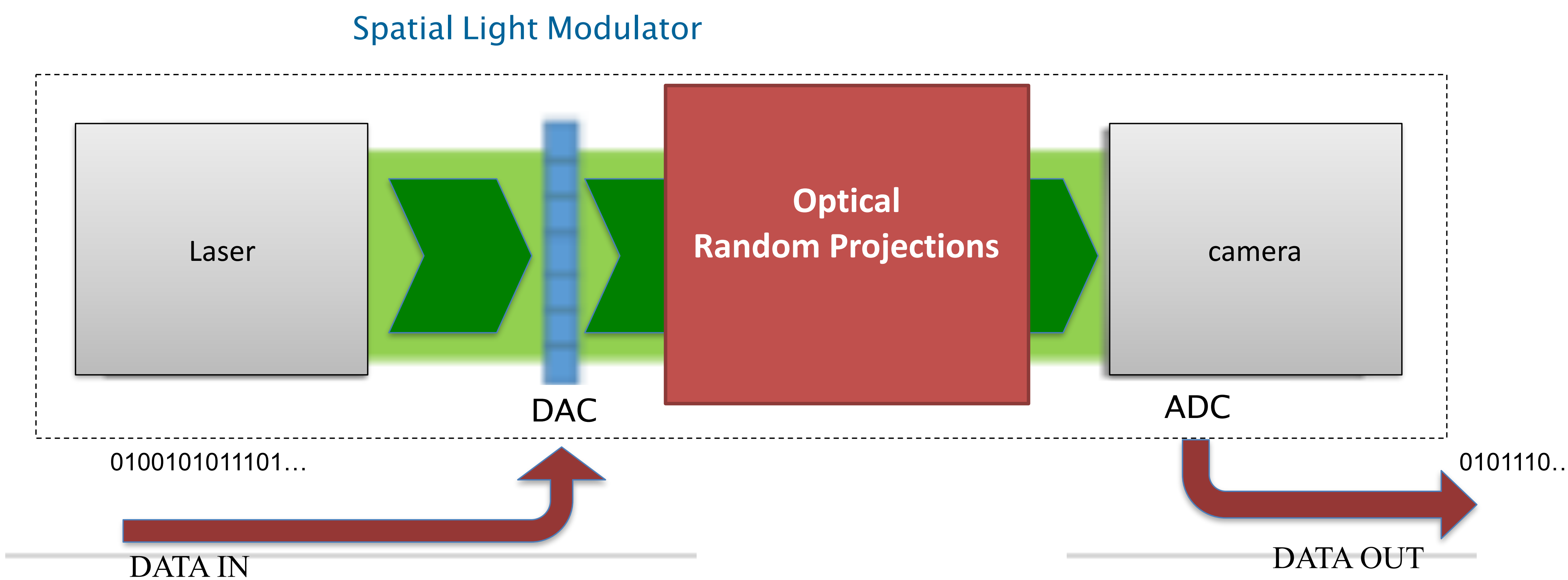
- … get Superman vision
- … learn from the blur
- … make pythons crawl faster

Towards optical computing



Towards optical computing

Now, let us just only consider the previous experiment as a "black box" with input in the SLM and output on the CCD



Towards optical computing

This performs in the analog domain

$$y = |Mx|^2$$

with M a *complex* random iid matrix

« **Random Projections** »

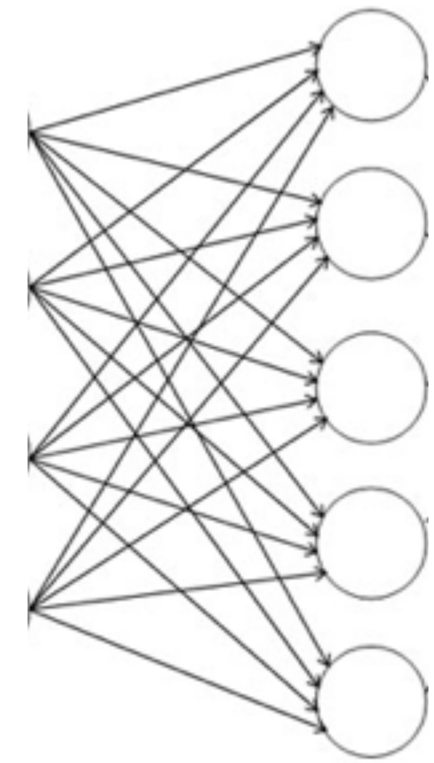
Very Large **&** **Fast**

size higher than
$10^6 \times 10^6$
(TBs of memory)

kHz operation
$\rightarrow 10^3$ such
multiplies / s

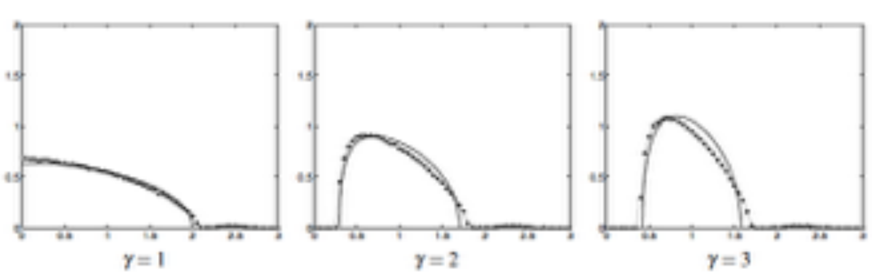
Equivalent 10^{15} Operations / s : if it were a computer it would be in the PetaOPS range

Towards optical computing



- A matrix-vector multiplication followed by a non-linearity:
  a fully connected layer of a Neural Network

- Fixed dense random weights - you can guarantee their distribution (Gaussian iid complex)



Marčenko-Pastur law
on singular values

- Random projections made **O(n^2)→ O(1)**

What does it enable ?

Three case studies

1/ Simple proof-of-concept of image classification based on **Kernel Ridge Regression**, where the random features are obtained with the optical experiment.

2/ **Fast Transfer Learning**, on a VGG16 architecture

3/ Optical **Echo-State Network**

Case study 1: classification with kernel ridge regression

training

U : data   Y: labels

$$\underset{\beta \in \mathbb{R}^{p \times q}}{\operatorname{argmin}} ||\mathbf{U}\beta - \mathbf{Y}||_2^2 + \gamma||\beta||_2^2$$

<u>Example</u> : classifying the MNIST database

training set of 60000 training pictures (28x28) of handwritten digits

test set of 10000 digits

regression

$$\beta = (\mathbf{U}^T\mathbf{U} + \gamma \mathbf{I}_p)^{-1}\mathbf{U}^T\mathbf{Y} = \mathbf{U}^T(\mathbf{U}\mathbf{U}^T + \gamma \mathbf{I}_n)^{-1}\mathbf{Y}$$

$$\tilde{\mathbf{Y}} = \tilde{\mathbf{U}}\beta = \tilde{\mathbf{U}}(\mathbf{U}^T\mathbf{U} + \gamma \mathbf{I}_p)^{-1}\mathbf{U}^T\mathbf{Y}$$

$$= \tilde{\mathbf{U}}\mathbf{U}^T(\mathbf{U}\mathbf{U}^T + \gamma \mathbf{I}_n)^{-1}\mathbf{Y}$$

These are inner products

inverting this N x N matrix can be hard

Case study 1: classification with kernel ridge regression

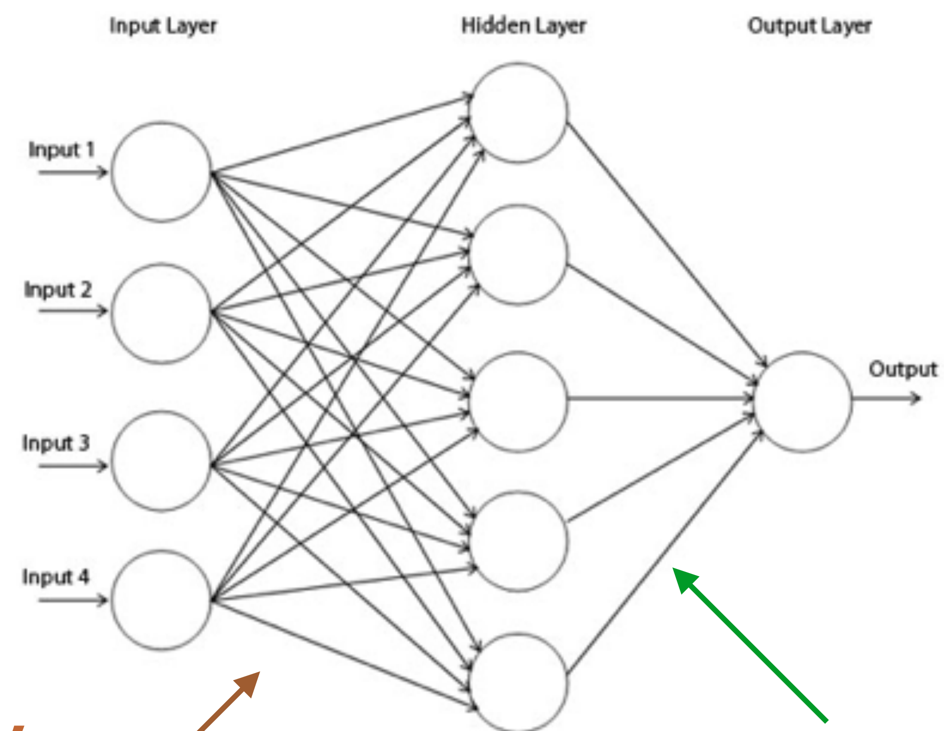
LightOn

Using random features

In the spirit of Rahimi-Recht :

$$\mathbf{X}_{i,j} = \phi((\mathbf{W}\mathbf{U}_i)_j + \mathbf{b}_j)$$

W random complex matrix with gaussian i.i.d. entries, b bias, and ϕ a non-linearity



Fully connected layer with fixed random weights

Hidden layer with *n* random features

Ridge regression on output

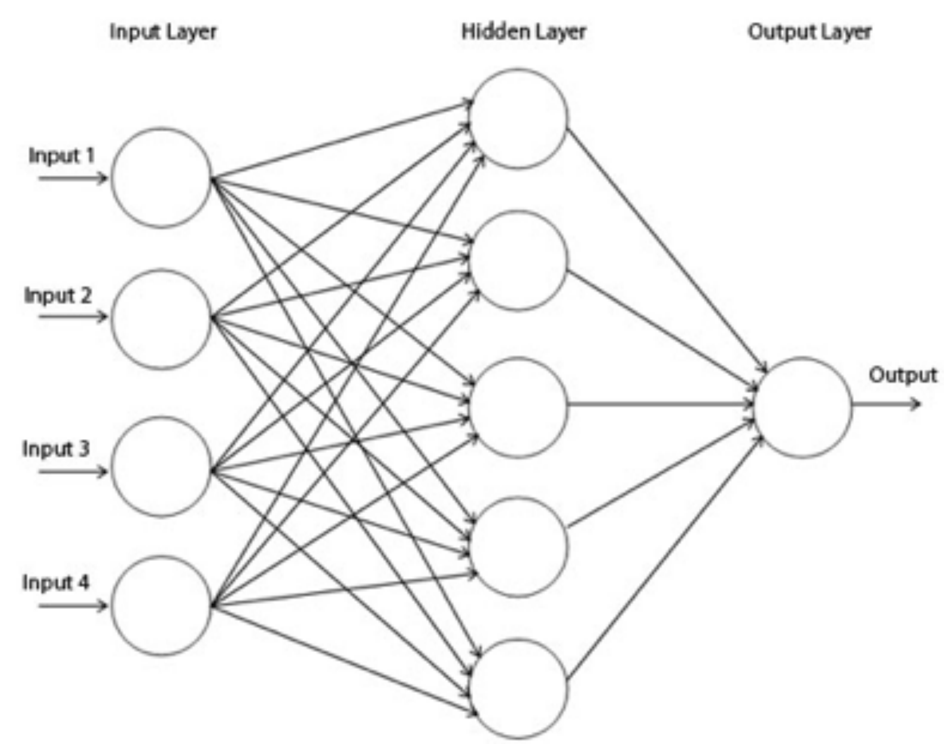
Case study 1: classification with kernel ridge regression

Using random features

In the spirit of Rahimi-Recht :

$$\mathbf{X}_{i,j} = \phi((\mathbf{W}\mathbf{U}_i)_j + \mathbf{b}_j)$$

W random complex matrix with gaussian i.i.d. entries, b bias, and ϕ a non-linearity



$$\tilde{\mathbf{Y}} = \tilde{\mathbf{X}}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T + \gamma\mathbf{I}_n)^{-1}\mathbf{Y} = \tilde{\mathbf{X}}(\mathbf{X}^T\mathbf{X} + \gamma\mathbf{I}_N)^{-1}\mathbf{X}^T\mathbf{Y}$$

of size N x N

N : number of training examples

of size ***n*** x ***n***

n number of random features
no dependency on N !

Case study 1: classification with kernel ridge regression

Kernel ridge regression

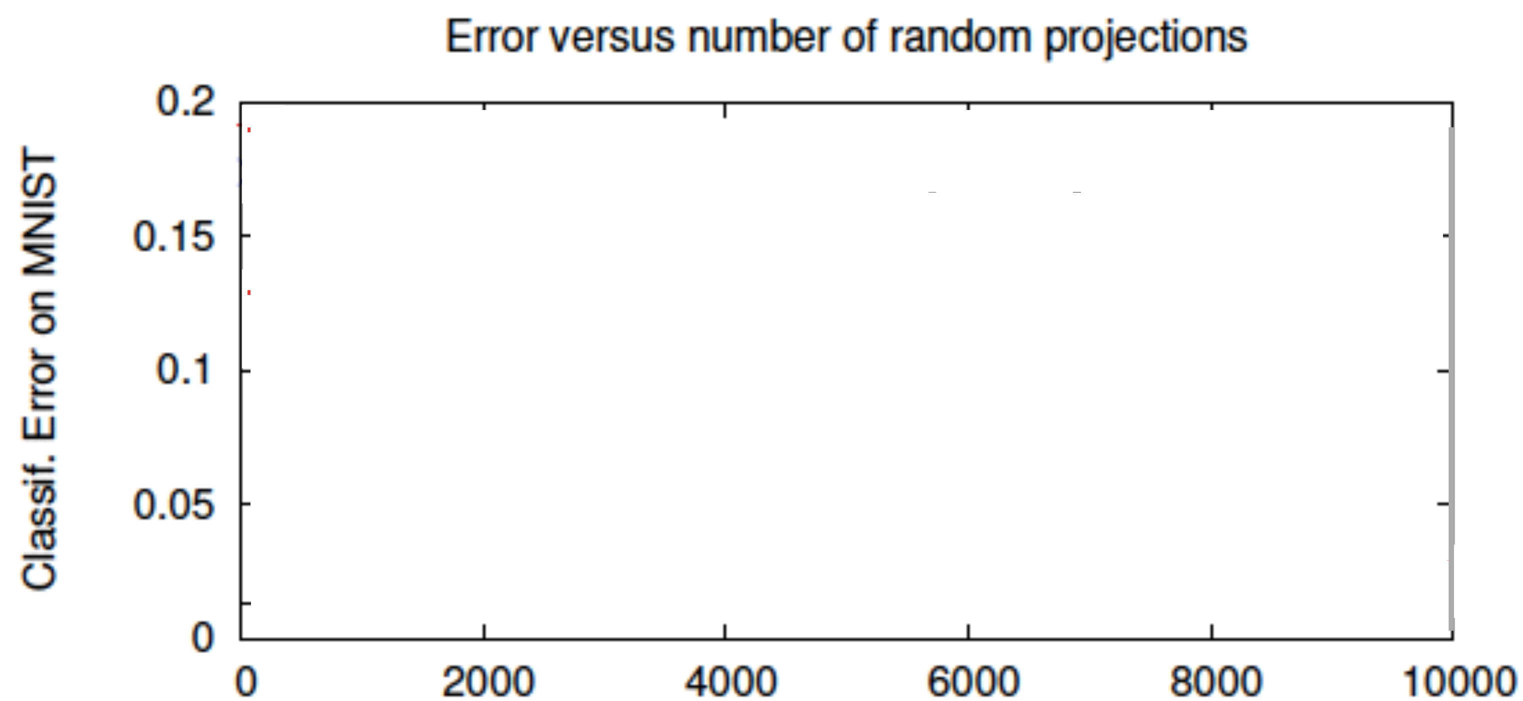
As n → ∞, inner products tend towards a **kernel** that can be computed explicitly

$$k(\mathbf{U}_i, \mathbf{U}_j) = \frac{\sqrt{\mathbf{U}_i^T\mathbf{U}_i\,\mathbf{U}_j^T\mathbf{U}_j}}{2}\left\{-(\sin^2\theta)\mathcal{E}_K\left[\cos^2\theta\right] + 2\mathcal{E}_E\left[\cos^2\theta\right] + |\sin\theta|\left(2\mathcal{E}_E\left[-\frac{\cos^2\theta}{\sin^2\theta}\right] - \mathcal{E}_K\left[-\frac{\cos^2\theta}{\sin^2\theta}\right]\right)\right\}$$

$\mathcal{E}_K[.]$ and $\mathcal{E}_E[.]$ are the complete elliptic integrals of the first / second kind
θ is the angle between $\mathrm{U}i$ and $\mathrm{U}j$

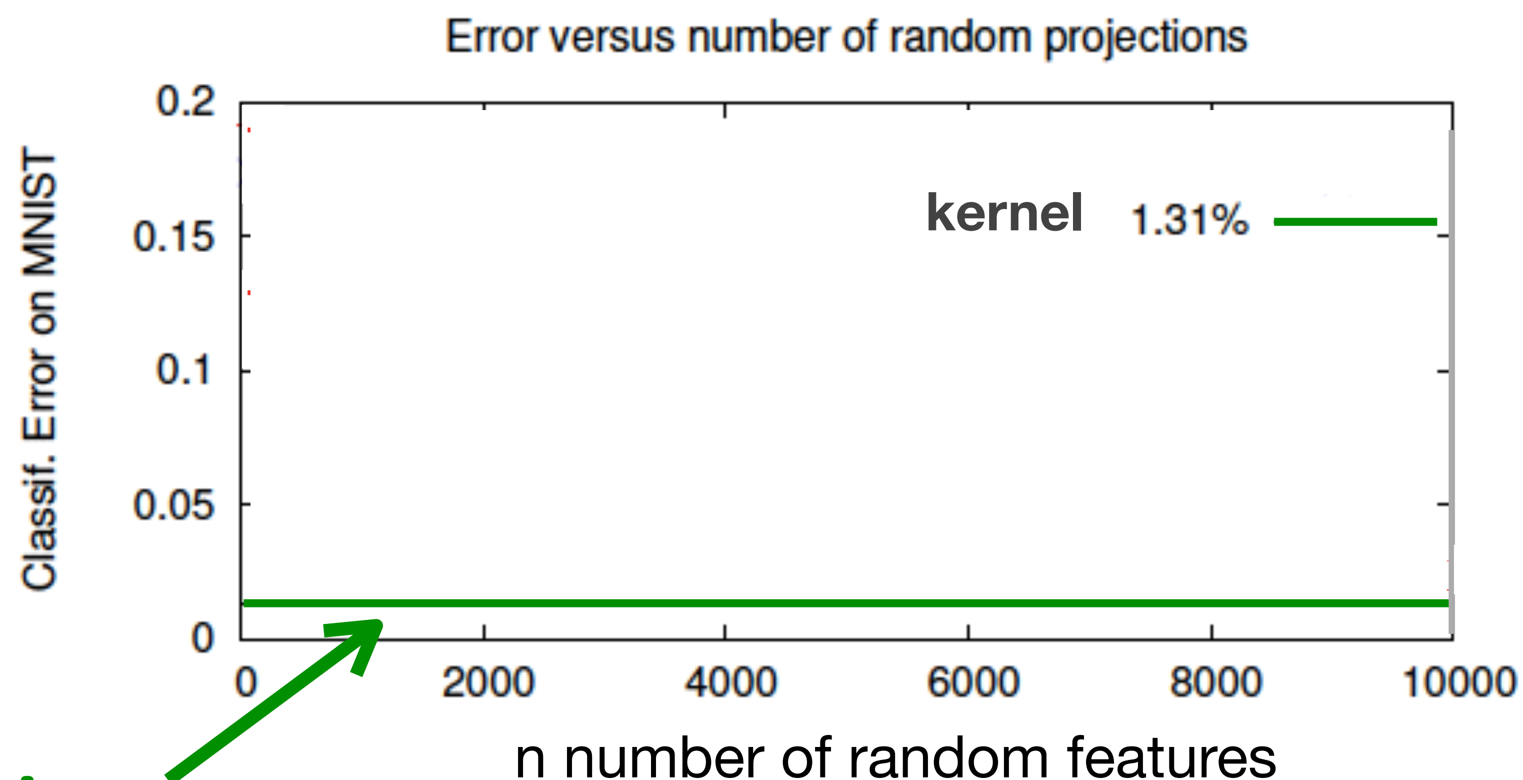
This kernel *numerically* provides a 1.31 % error rate on MNIST

Case study 1: classification with kernel ridge regression



n number of random features

Case study 1: classification with kernel ridge regression

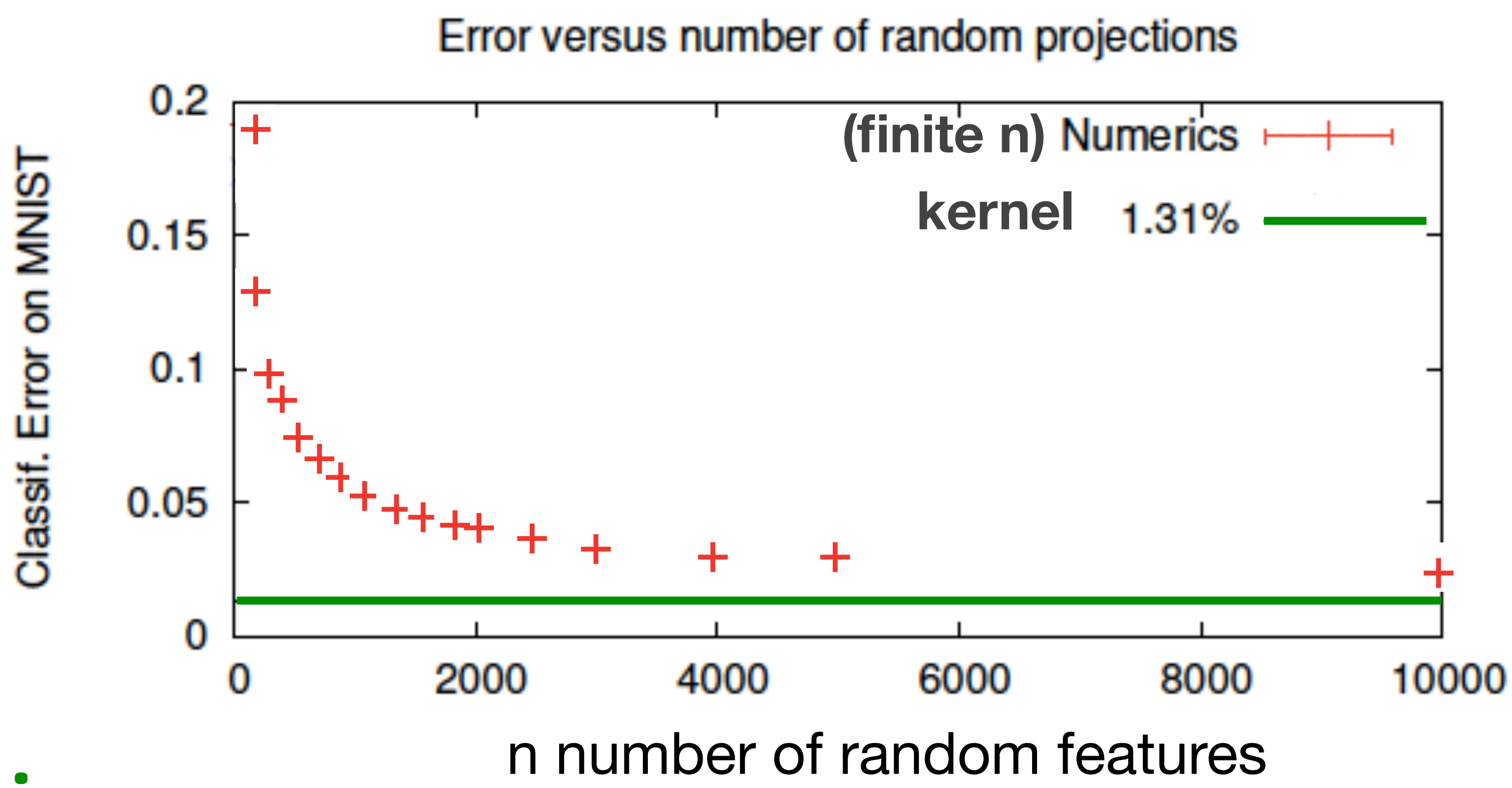


mathematics

kernel: asymptotic behavior as $n \to \infty$

Case study 1: classification with kernel ridge regression

numerical simulations

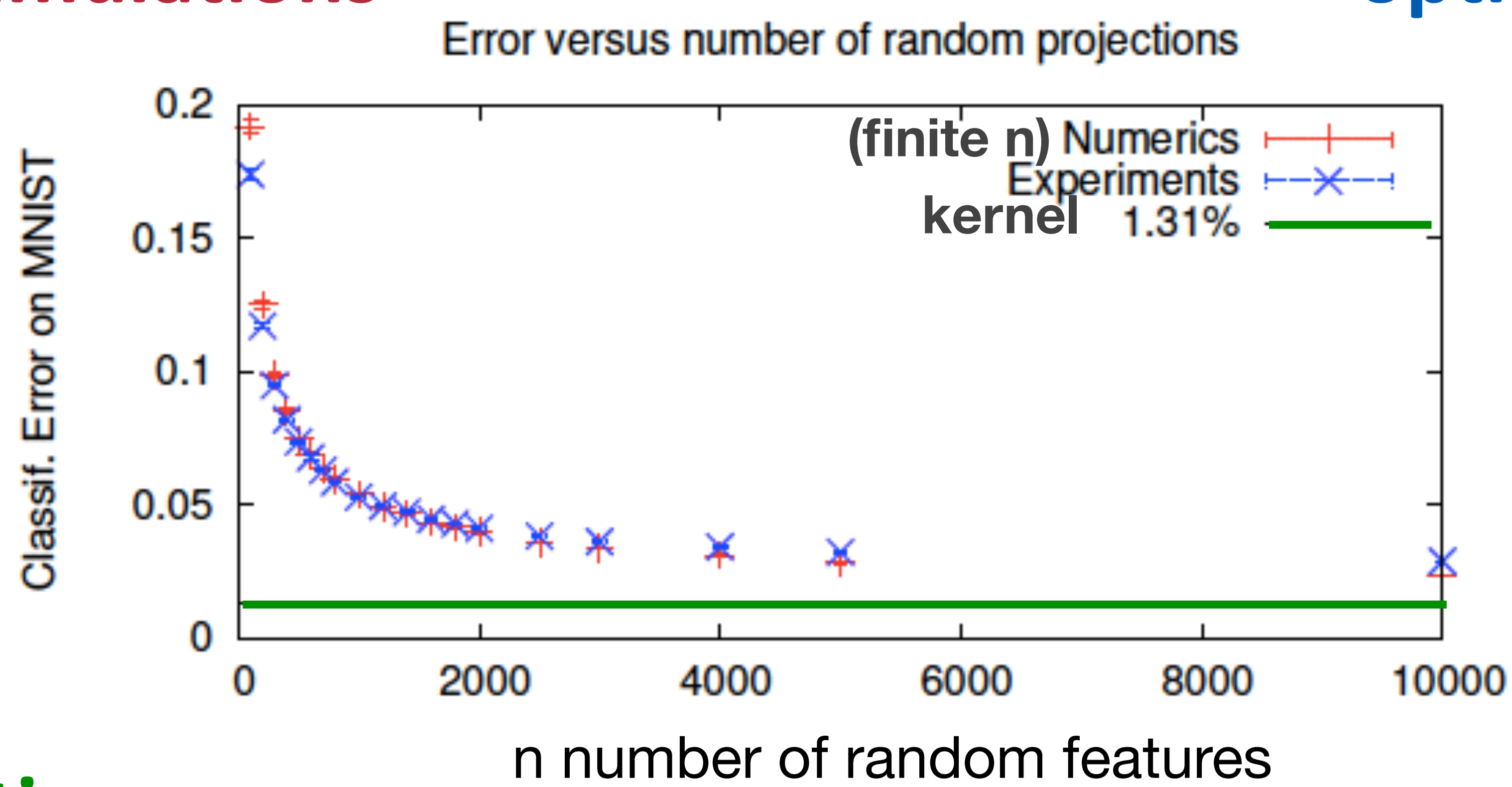


mathematics

Case study 1: classification with kernel ridge regression

numerical simulations

optics experiment



mathematics

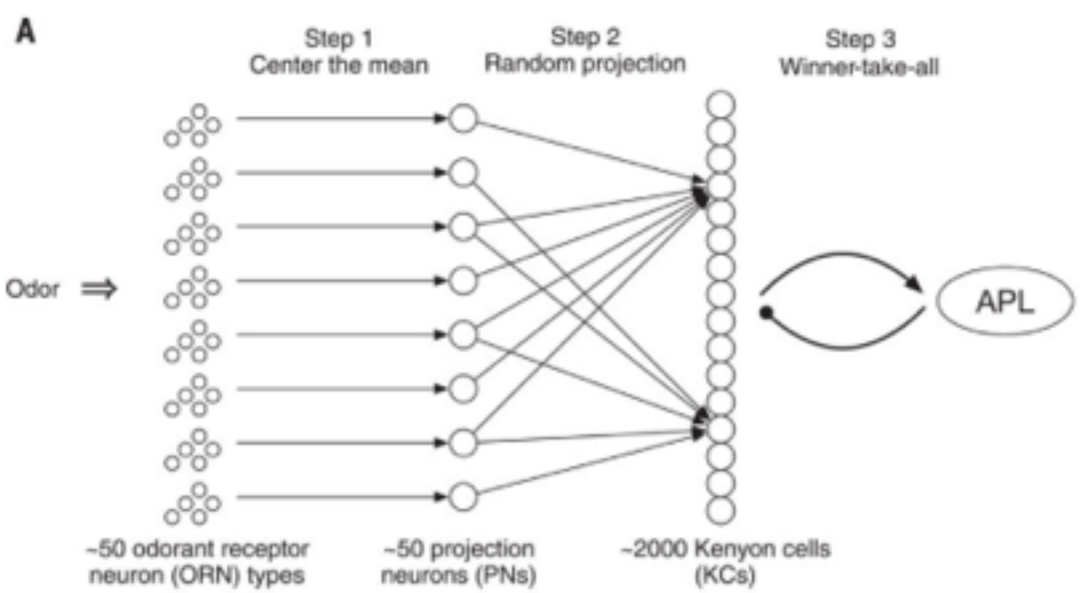
Biological motivation for dimensionality expansion with LSH

Science
Vol 358, Issue 6364
10 November 2017

Fly brain inspires computing algorithm

Flies use an algorithmic neuronal strategy to sense and categorize odors. Dasgupta *et al.* applied insights from the fly system to come up with a solution to a computer science problem. On the basis of the algorithm that flies use to tag an odor and categorize similar ones, the authors generated a new solution to the nearest-neighbor search problem that underlies tasks such as searching for similar images on the web.

Muhammad M. Karim, GDFL 1.2



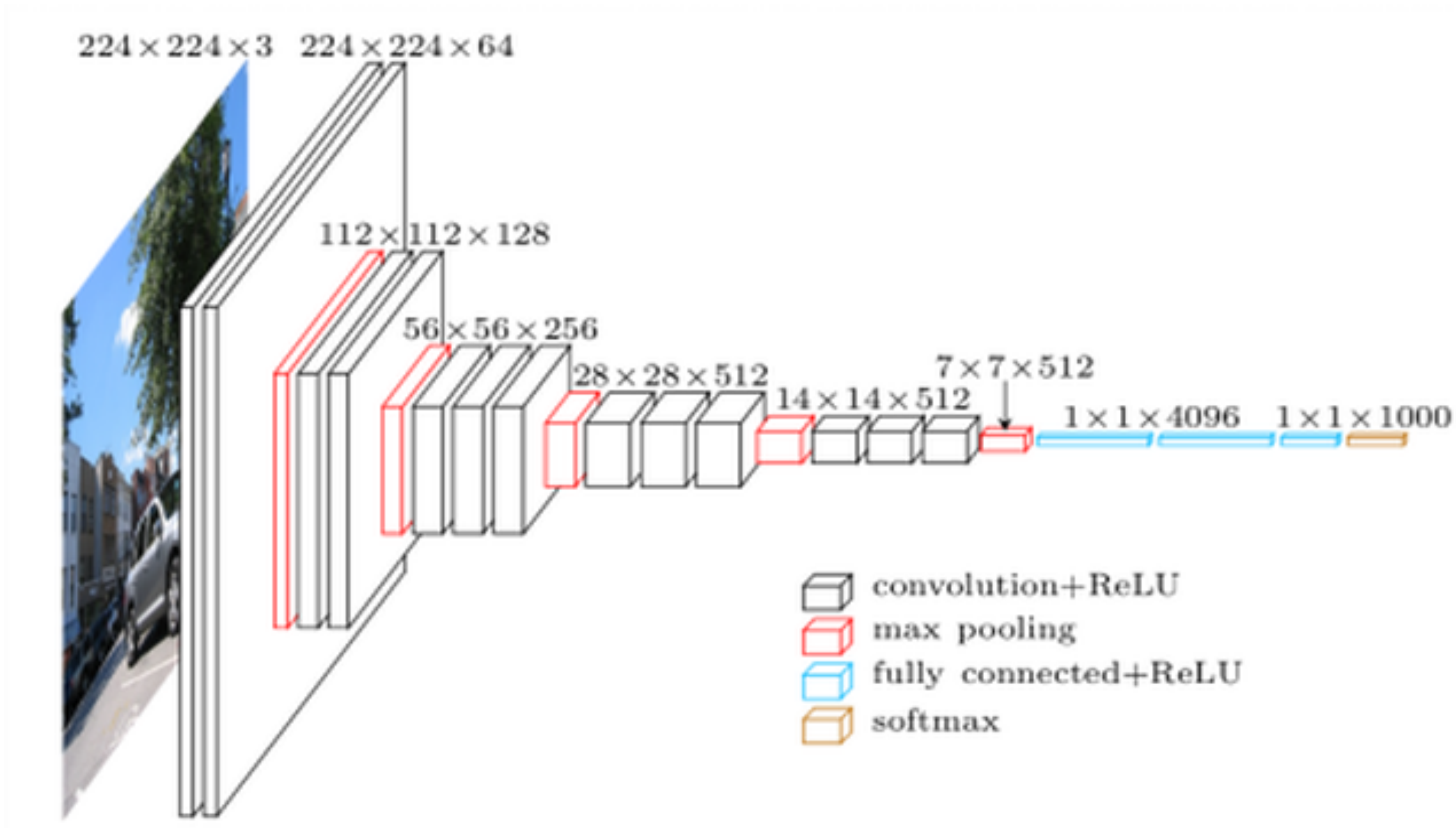
A neural algorithm for a fundamental computing problem

Sanjoy Dasgupta[1], Charles F. Stevens[2,3], Saket Navlakha[4,*]
+ See all authors and affiliations

Science 10 Nov 2017:
Vol. 358, Issue 6364, pp. 793-796
DOI: 10.1126/science.aam9868

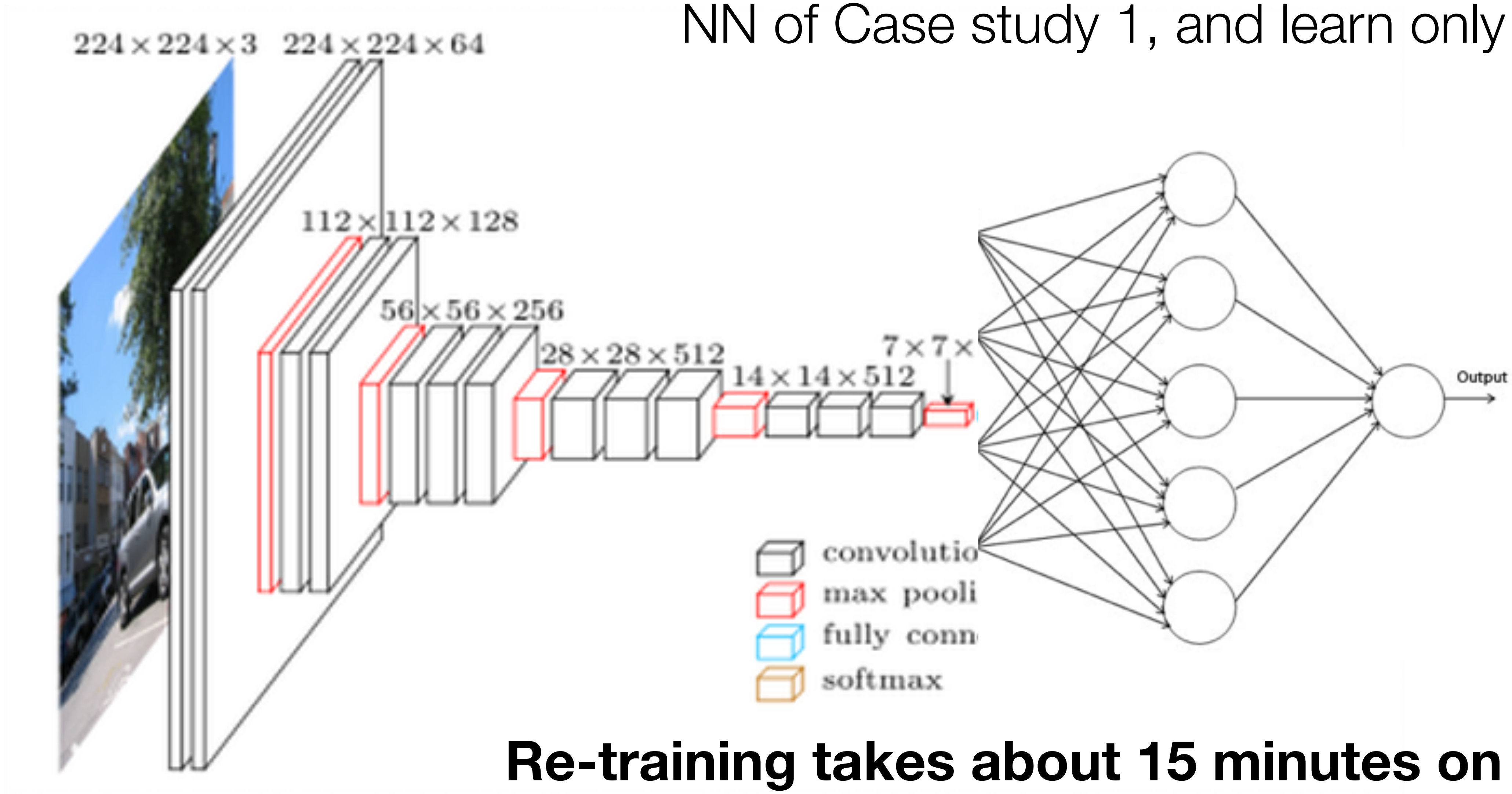
Case study 2: Fast Transfer Learning

- Start with a standard VGG16 [Simonyan & Zisserman '14] architecture
  - Train for a week on ImageNet with a good GPU



Case study 2: Fast Transfer Learning

- Now comes a second dataset : STL10
  - Keep trained convolutional layers unchanged
  - Replace Fully Connected + ReLU by the 2-layer NN of Case study 1, and learn only last layer

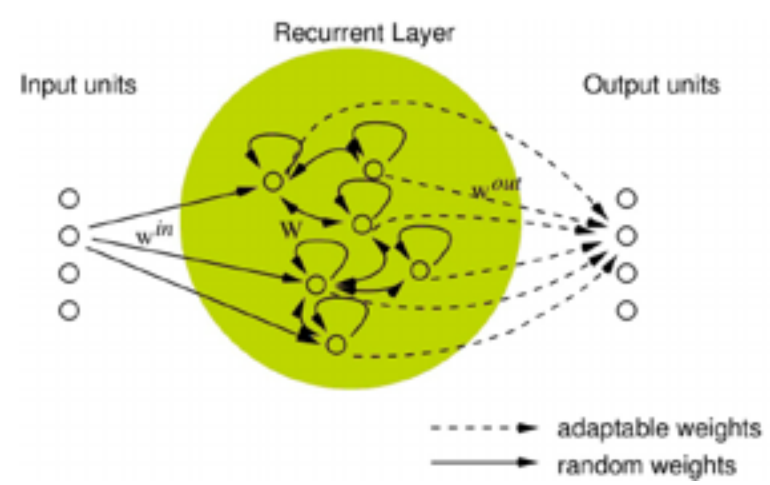


Re-training takes about 15 minutes on a good GPU

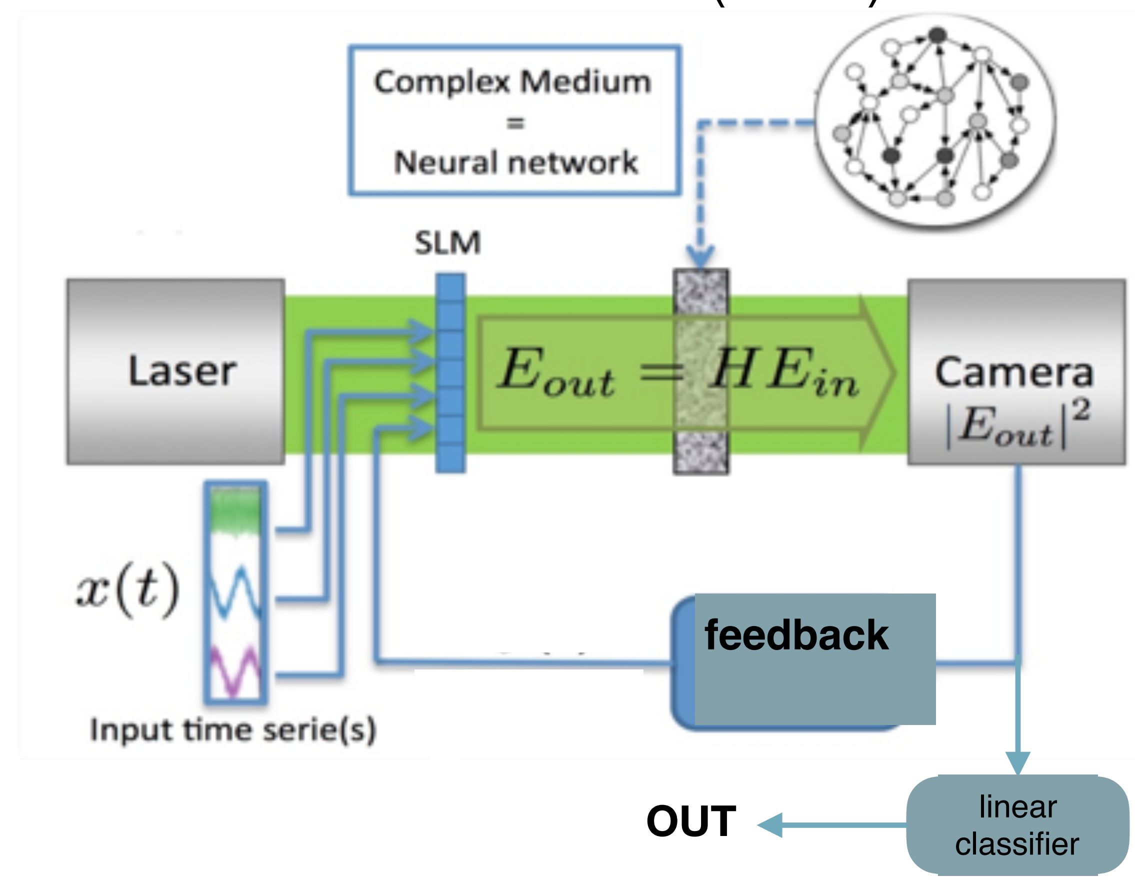
LightOn

Case study 3: Optical Echo-State Networks

A physical implementation of large-scale echo-state networks (ESN)

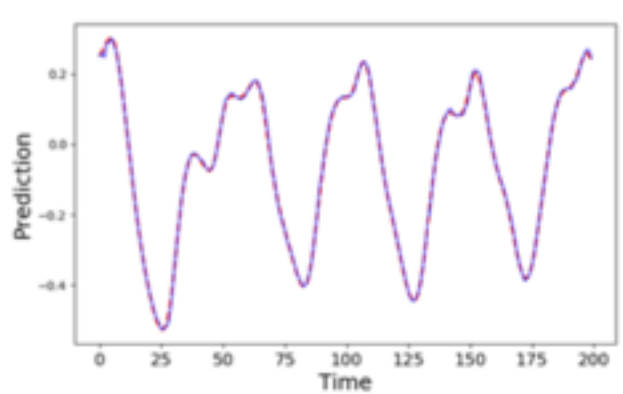


[diagram from Obst et al. 2013]



regression on complex time series

Ex: predict dynamics of Mackey-Glass eqs. *(Dong. et al)*



2 orders of magnitude larger / x200 faster than standard PCs

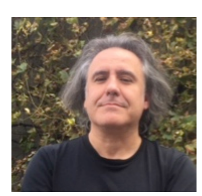
Outline

How to

- … get Superman vision
- … learn from the blur
- … make pythons crawl faster

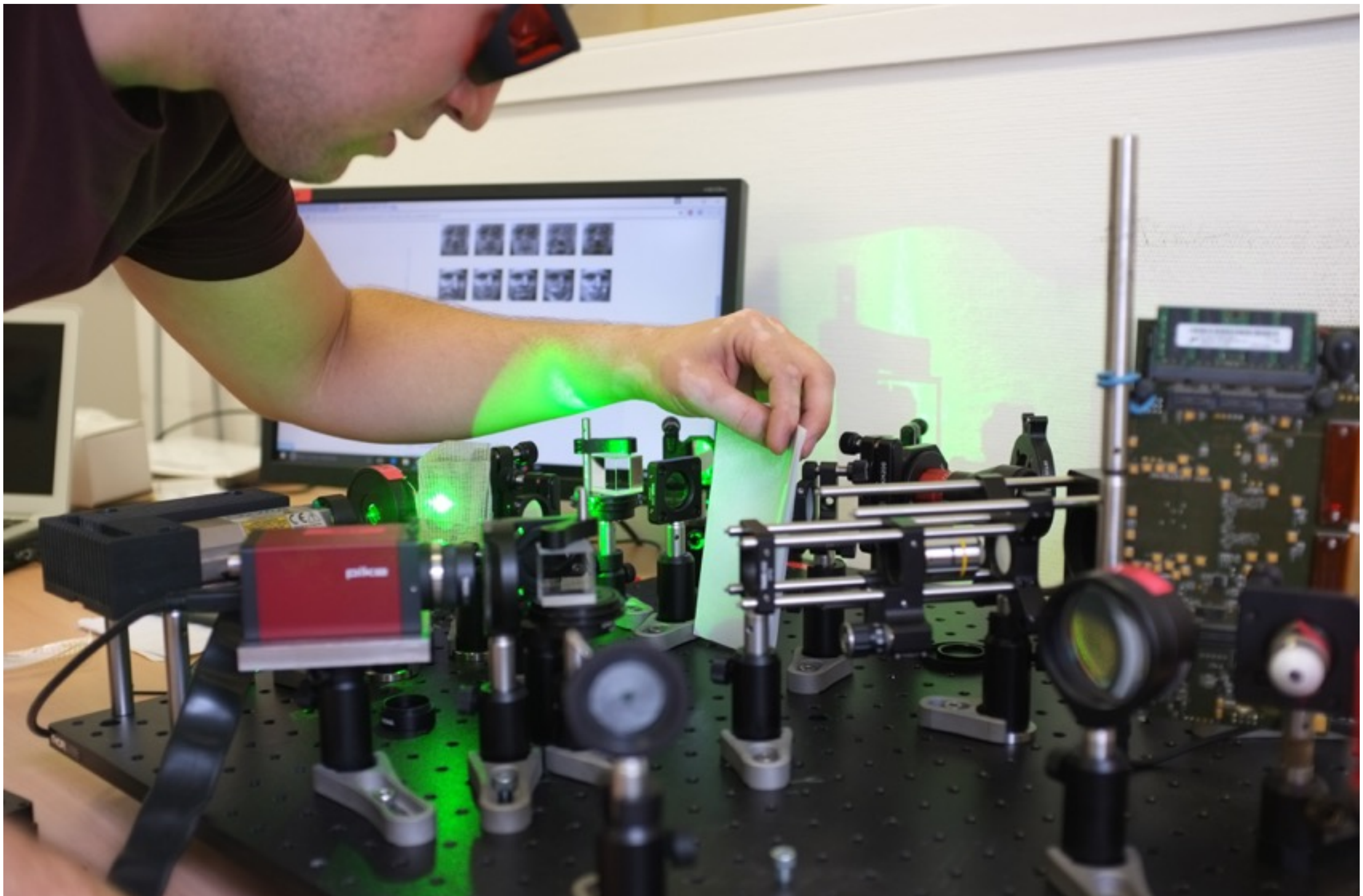
Technology Roadmap

- Created 2016
- 4 co-founders



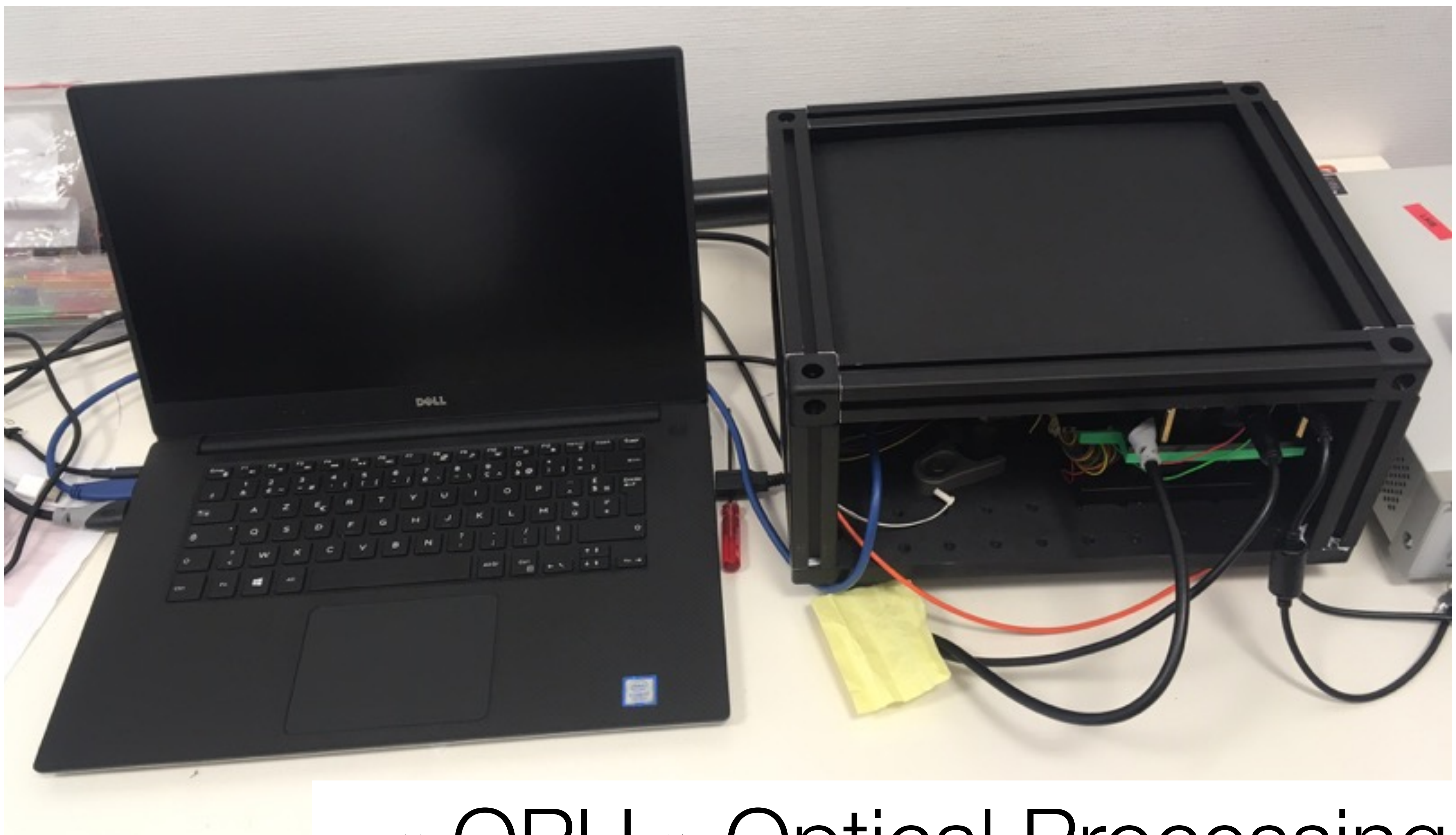

- 5 R&D engineers
- Based in Paris « Quartier Latin »

From lab experiment to prototype



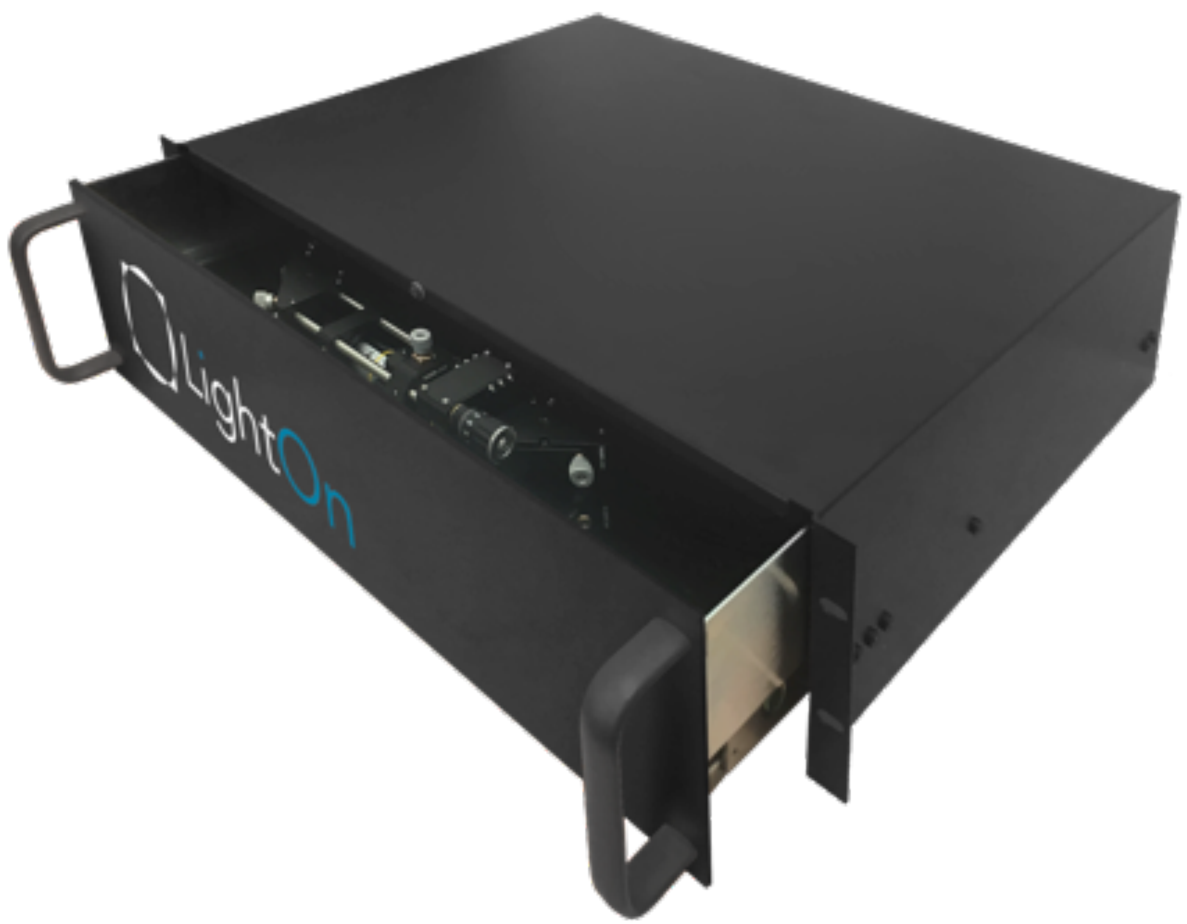
From lab experiment to prototype

Using only off-the-shelf components - First prototype Spring 2017



« OPU » Optical Processing Unit

Current OPU prototype



Rack-size OPU

low power (< 30W)

LightOn Cloud

- **OPU + CPU/GPU in an external datacenter**
- **Cloud service already operational** currently under alpha testing
- **Platform-as-a-Service** with integration within popular ML frameworks (Python-based: SciKit-Learn, TensorFlow in progress …)
- Available for beta-users Q2 2018 (VMs via OpenStack)

Take-home message Part 2

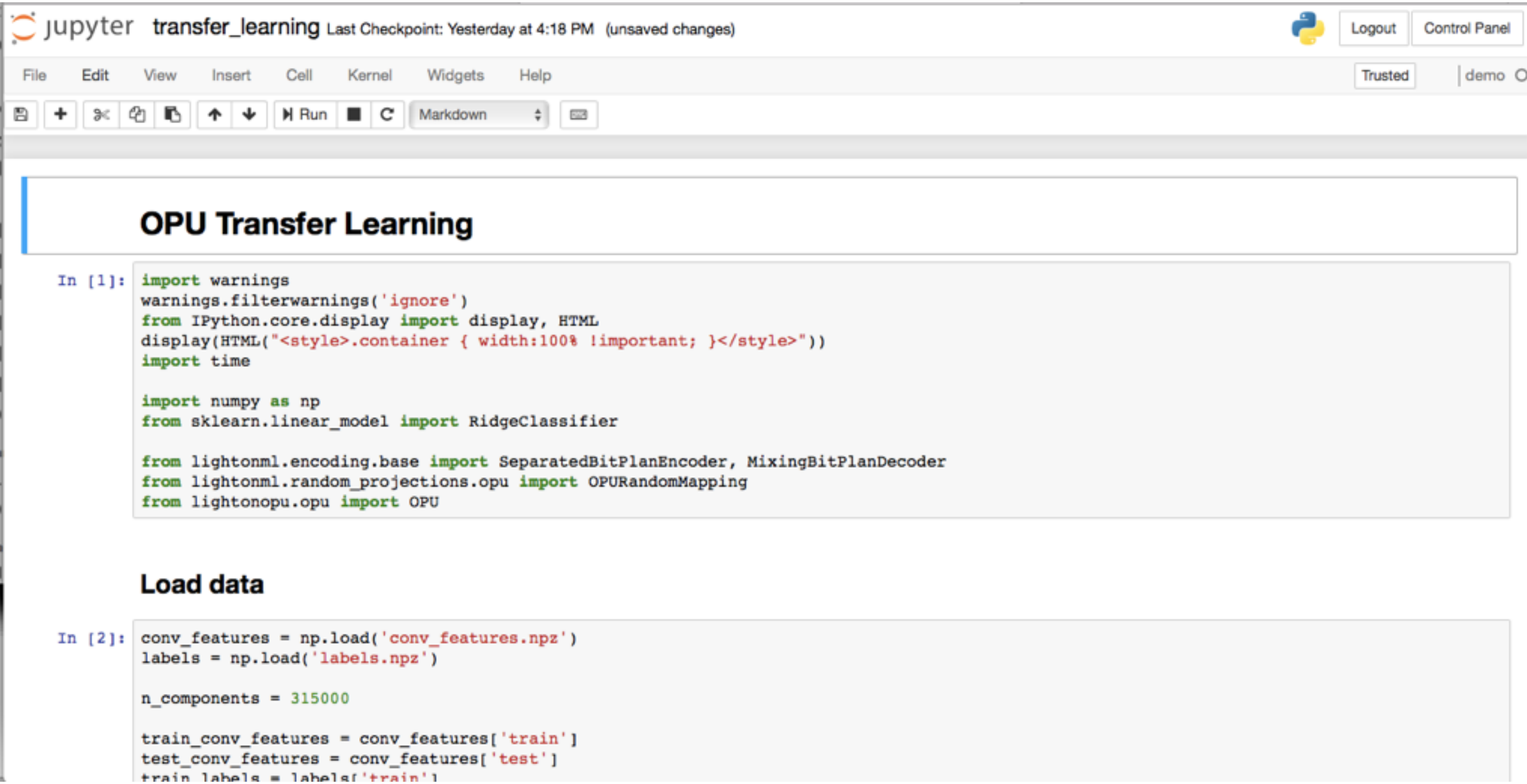
- **Three case studies** so far for the OPU
  - Kernel Ridge Regression
  - Echo-State Network
  - Fast Transfer Learning
- **More to come soon**
  - sketching distributions
  - dimensionality reduction for unsupervised learning
  - locality sensitive hashing for fast NN search
  - …

NEWMA: a new method for scalable model-free online change-point detection, Nicolas Keriven, Damien Garreau, Iacopo Poli, arXiv:1805.08061

- **What's your case study ?**
  - Register for beta test at http://www.lighton.io/lighton-cloud

LightOn

User Interface: Python / Jupyter notebooks



Selected references

- "Imaging With Nature: Compressive Imaging Using a Multiply Scattering Medium”, A. Liutkus et al., *Scientific Reports 4 (july 2014)*
- "Reference-less measurement of the transmission matrix of a highly scattering material using a DMD and phase retrieval techniques", A. Drémeau et al., *Optics Express 23(9), 2015*
- "Random Projections through multiple optical scattering: Approximating kernels at the speed of light", A. Saade et al., *Proc. ICASSP (2016)*
- "Scaling up Echo-State Networks with multiple light scattering", J. Dong et al., *arXiv: 1609.05204*
- "NEWMA: a new method for scalable model-free online change-point detection", Nicolas Keriven, Damien Garreau, Iacopo Poli, *arXiv:1805.08061*