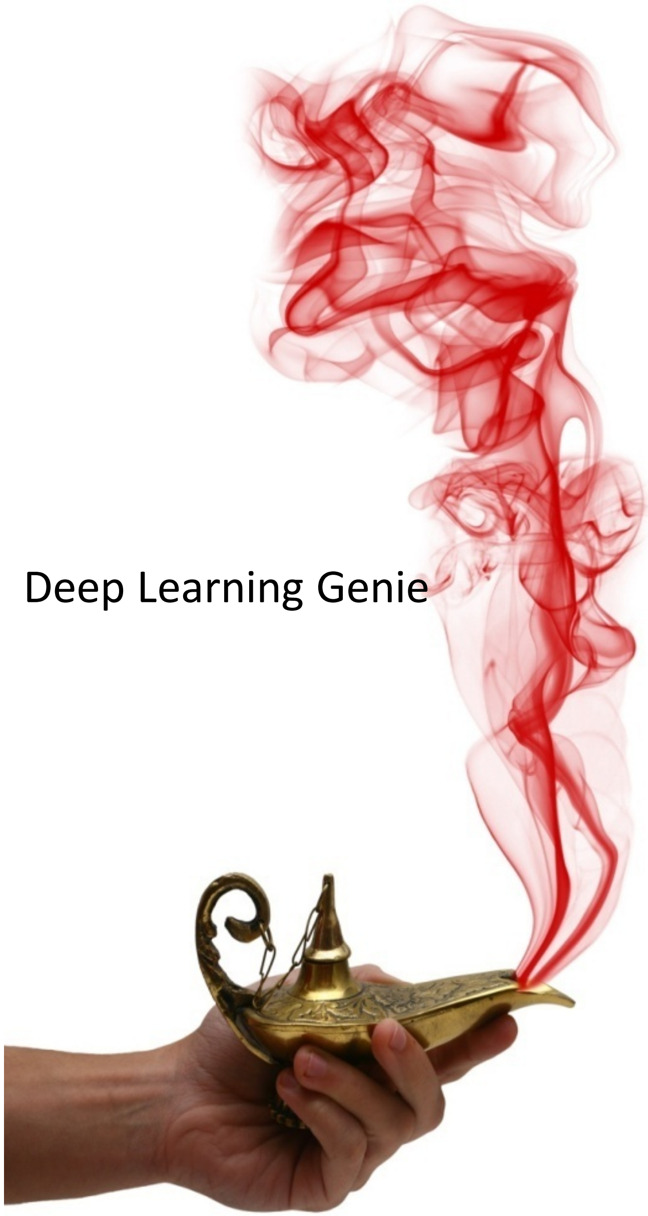
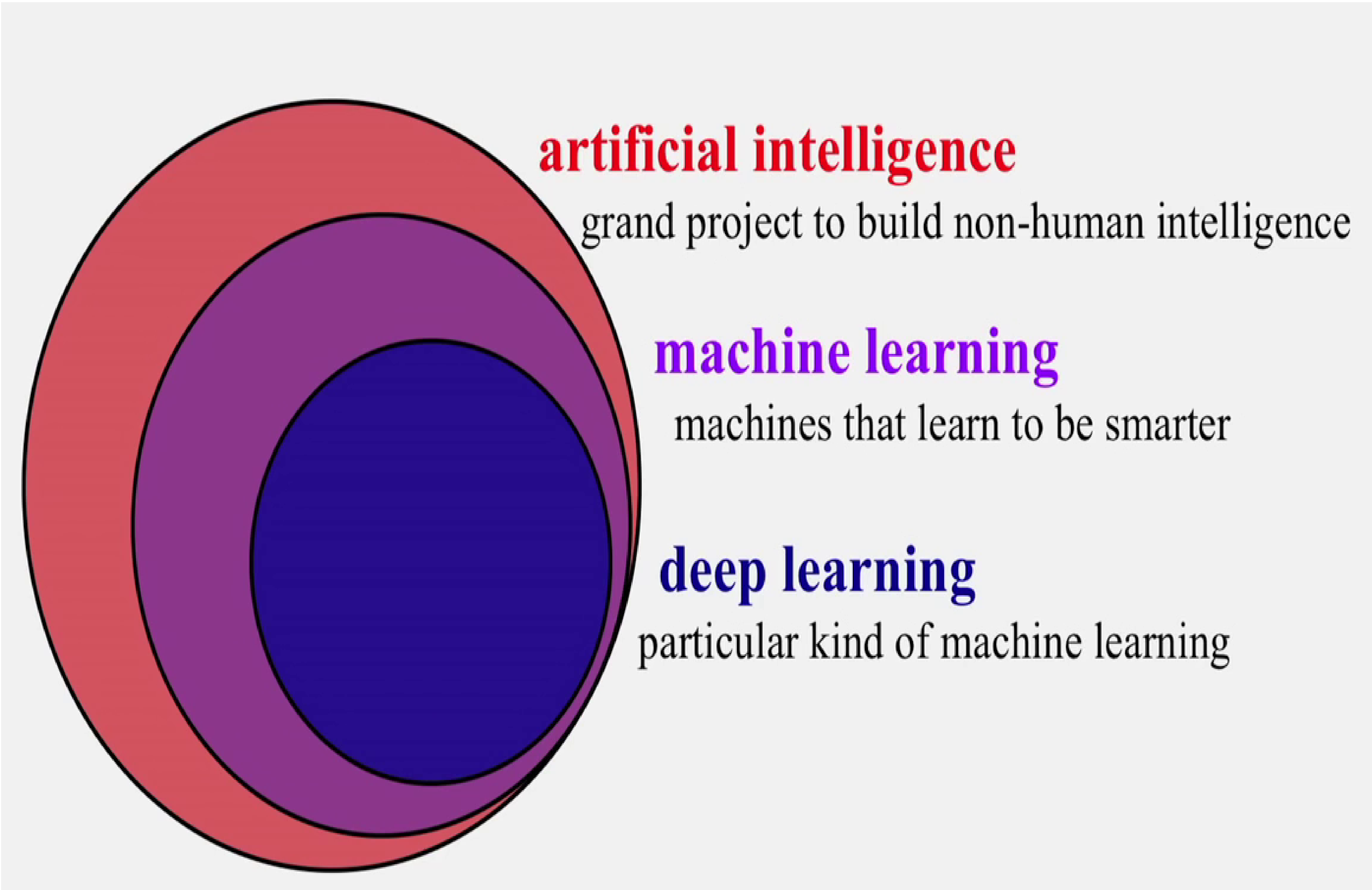
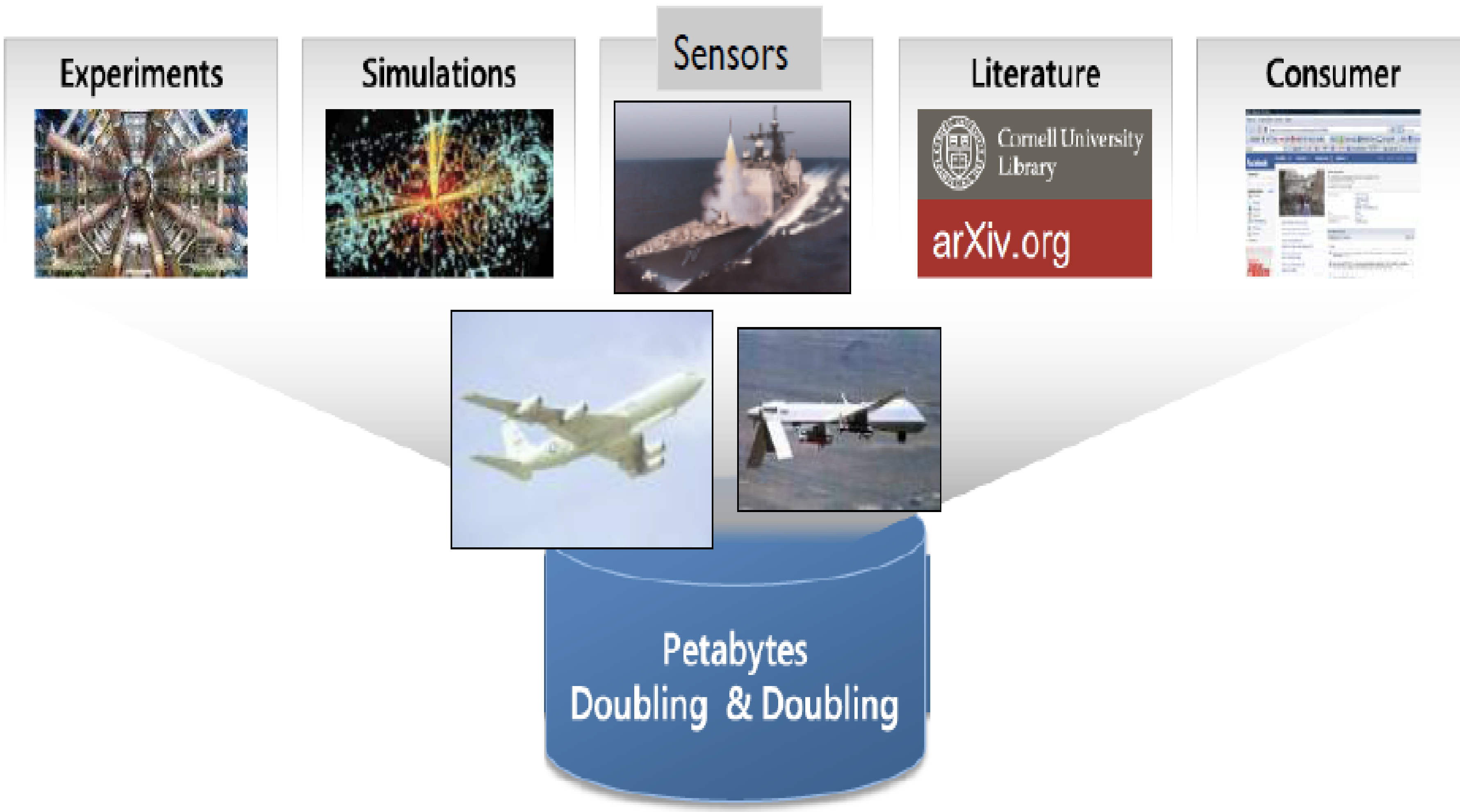


Machine learning applications in Fluid Dynamics

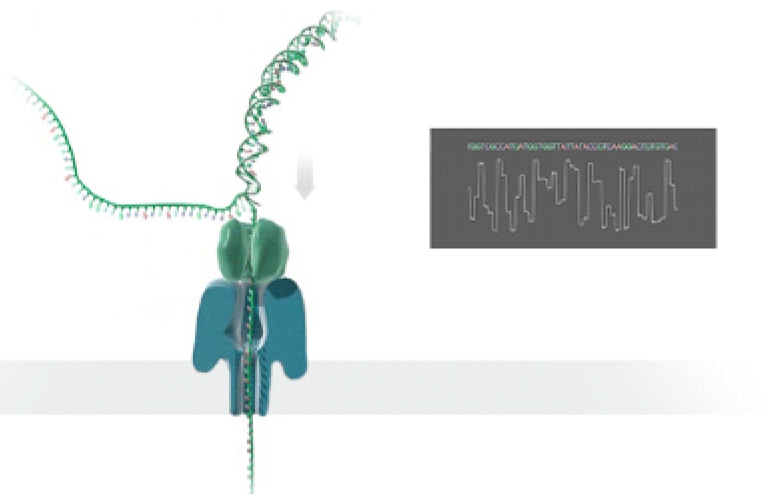
Soman K.P., E.A Gopalakrishnan
Centre for Computational Engineering
Amrita Vishwa Vidyapeetham


artificial intelligence
grand project to build non-human intelligence
machine learning
machines that learn to be smarter
deep learning
particular kind of machine learning

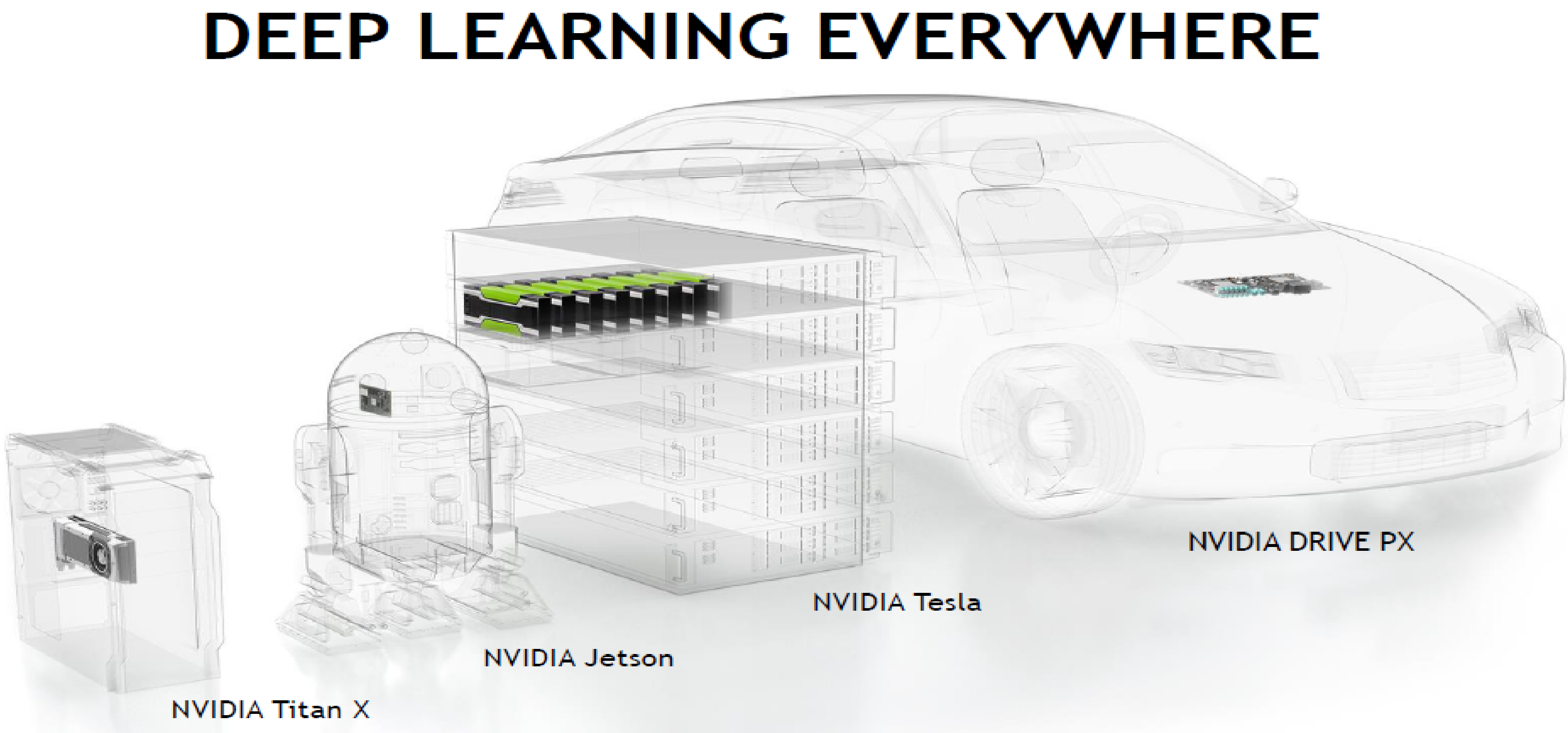
Explosion of Data Sources



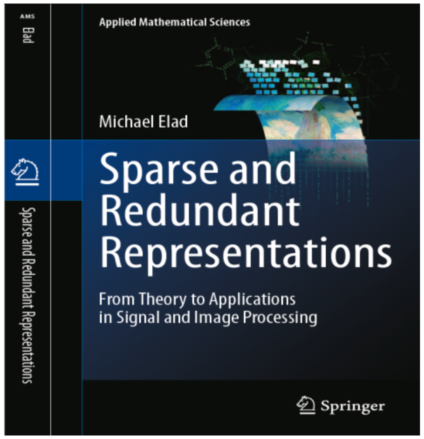
Desktop DNA/RNA Sequencer



Computing power for Deep Learning.



Deep Learning's Impact on Image Processing, Mathematics, and Humanity



By [Michael Elad]

Researchers developed beautiful and deep mathematical ideas with tools from **partial differential equations**, such as **anisotropic diffusion** and **total variation**, energy minimization viewpoint, adoption of a geometric interpretation of images as manifolds, use of the Beltrami flow, and more.

Harmonic analysis and approximation theory have also served the denoising task, leading to major breakthroughs with **wavelet theory and sparse representations**.

Other brilliant ideas included **low-rank approximation, non-local means, Bayesian estimation, and robust statistics**.

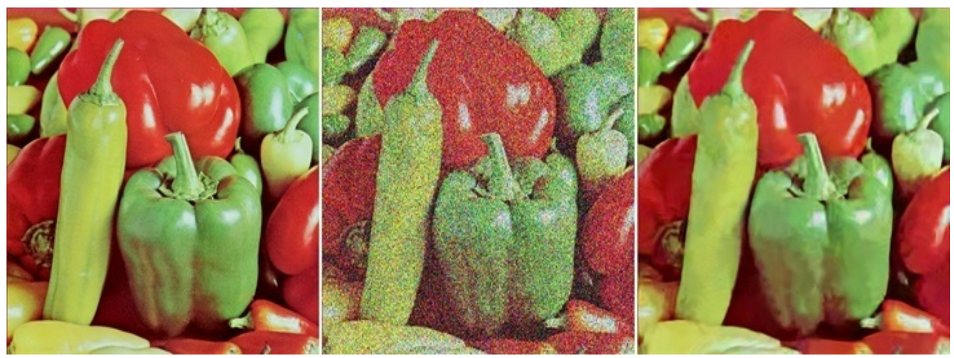
We have hence gained vast knowledge in image processing over the past three decades, impacting many other image processing tasks and effectively upgrading this field to be mathematically well-founded

But one day everything got washed away

Deep Learning's Impact on Image Processing, Mathematics, and Humanity

In 2012, Harold Burger, Christian Schuler, and Stefan Harmeling decided to throw deep learning into this problem

end result was a network that performed better than *any* known image denoising algorithm at that time.



The above is not an isolated story. Today, deep learning treats many other image processing needs, with unsurpassed results. This is true for single image **super-resolution, demosaicing, deblurring, segmentation, image annotation, and face recognition, among others**.

AI' s impact is going to be more than that of PC and Mobile

AI will help us be...

healthier

happier

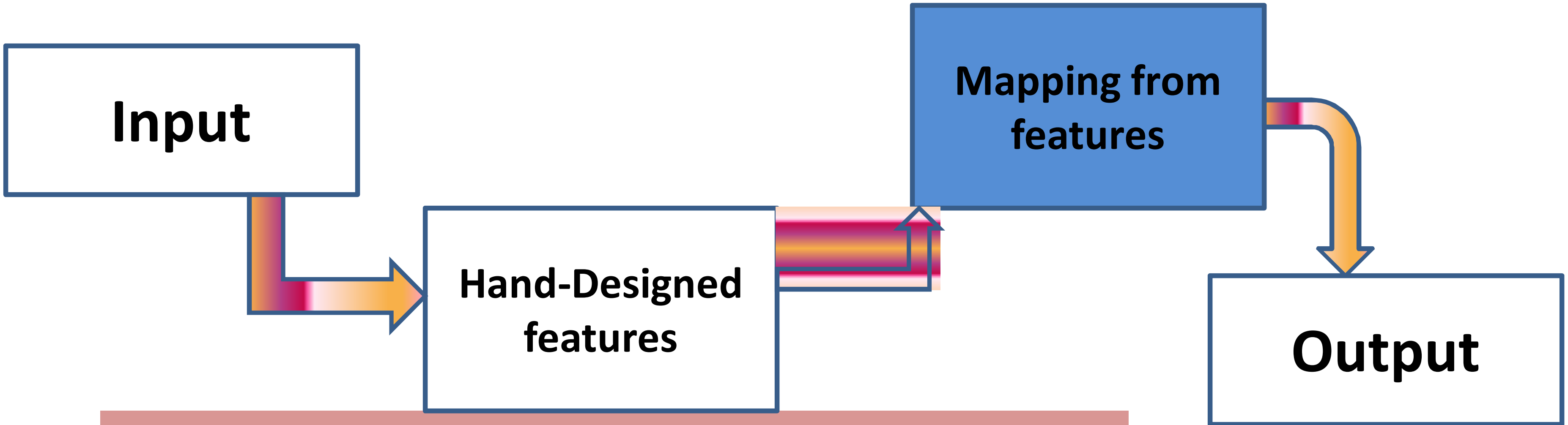
more productive

more creative

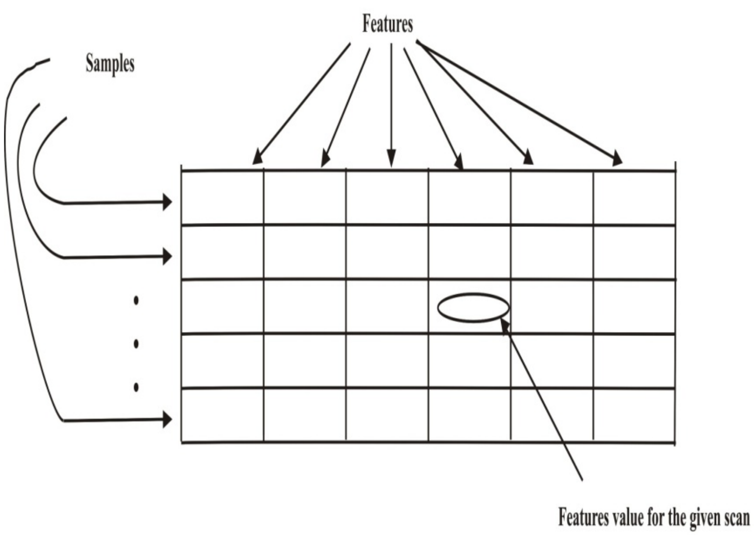
How do we learn?

We learn from examples and repeated practice

Classical machine learning



Acquire knowledge, by extracting patterns from raw data



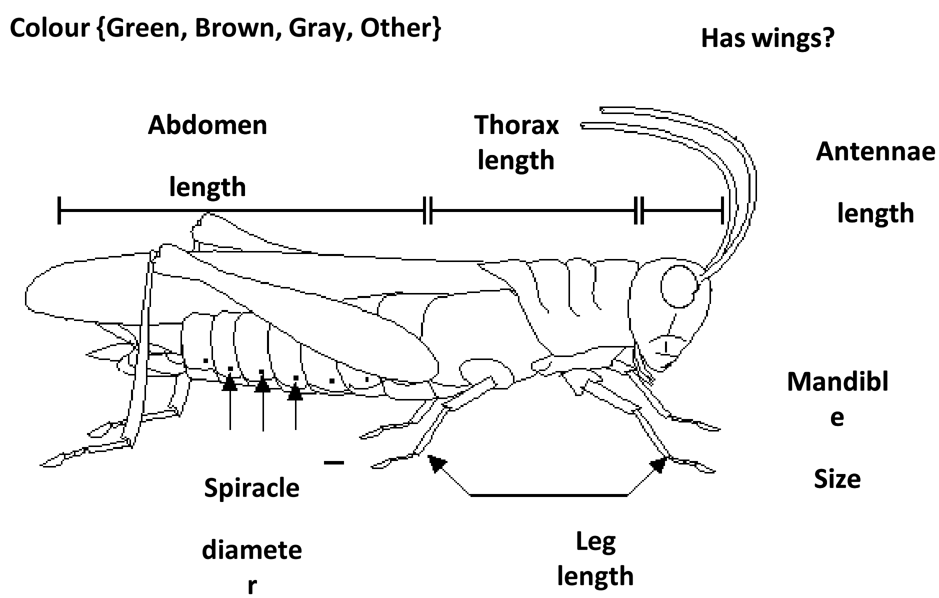


Table Medical Data

o	***Gender***	***Age***	***BP***	***Drug***
1	Male	20	Normal	A
2	Female	73	Normal	B
3	Male	37	High	A
4	Male	33	Low	B
5	Female	48	High	A
6	Male	29	Normal	A
7	Female	52	Normal	B
8	Male	42	Low	B
9	Male	61	Normal	B
10	Female	30	Normal	A
11	Female	26	Low	B
12	Male	54	High	A

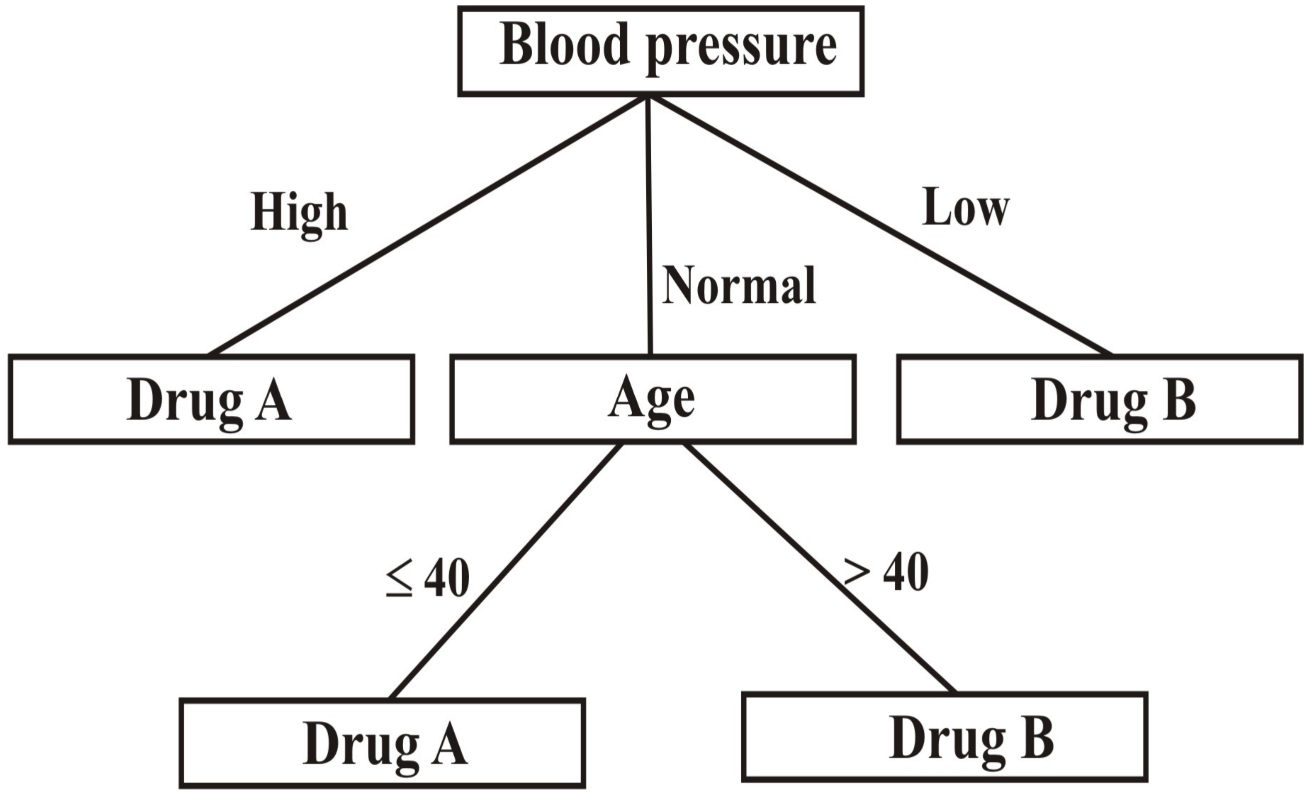


Fig .4.1 Decision Tree for the medical data.

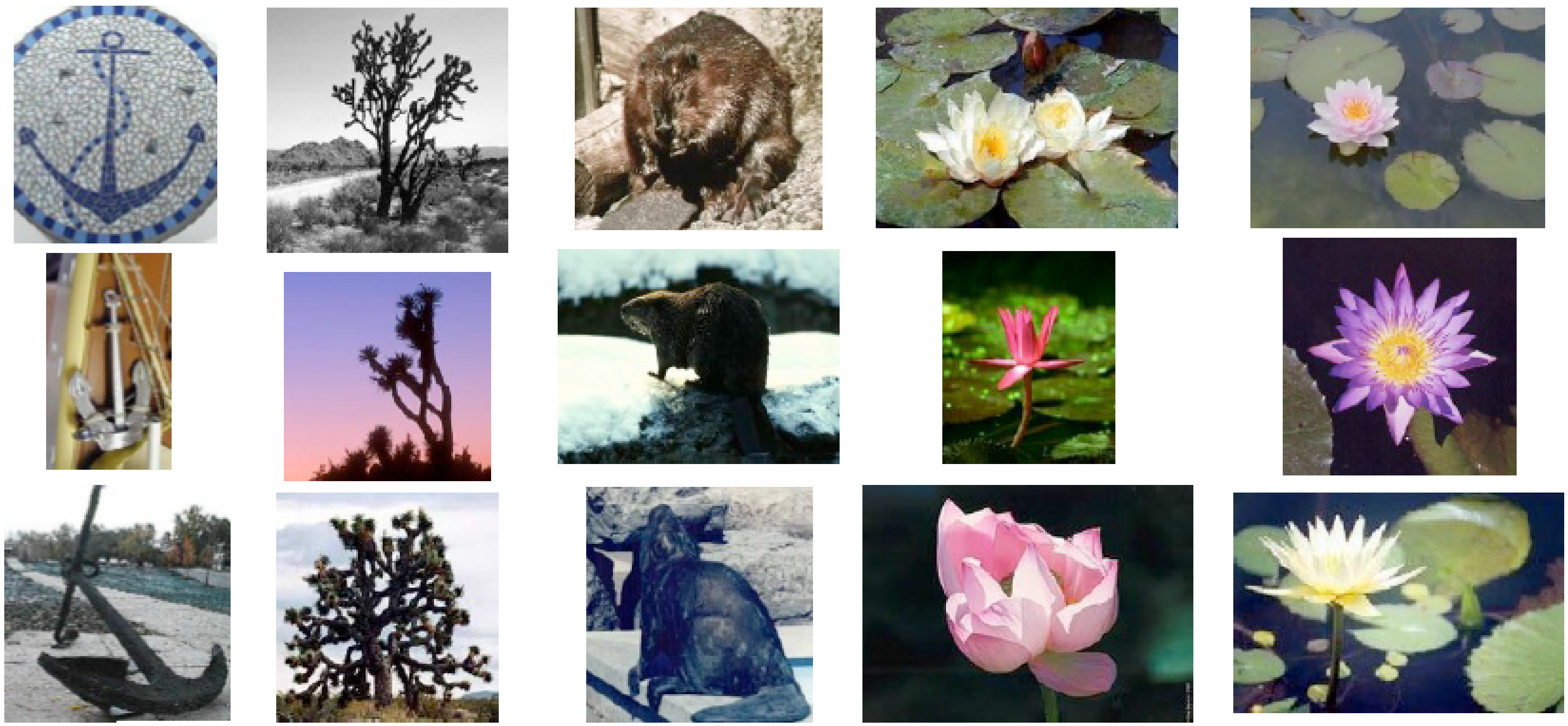
Problem with classical machine learning

Dealing with High dimensional data

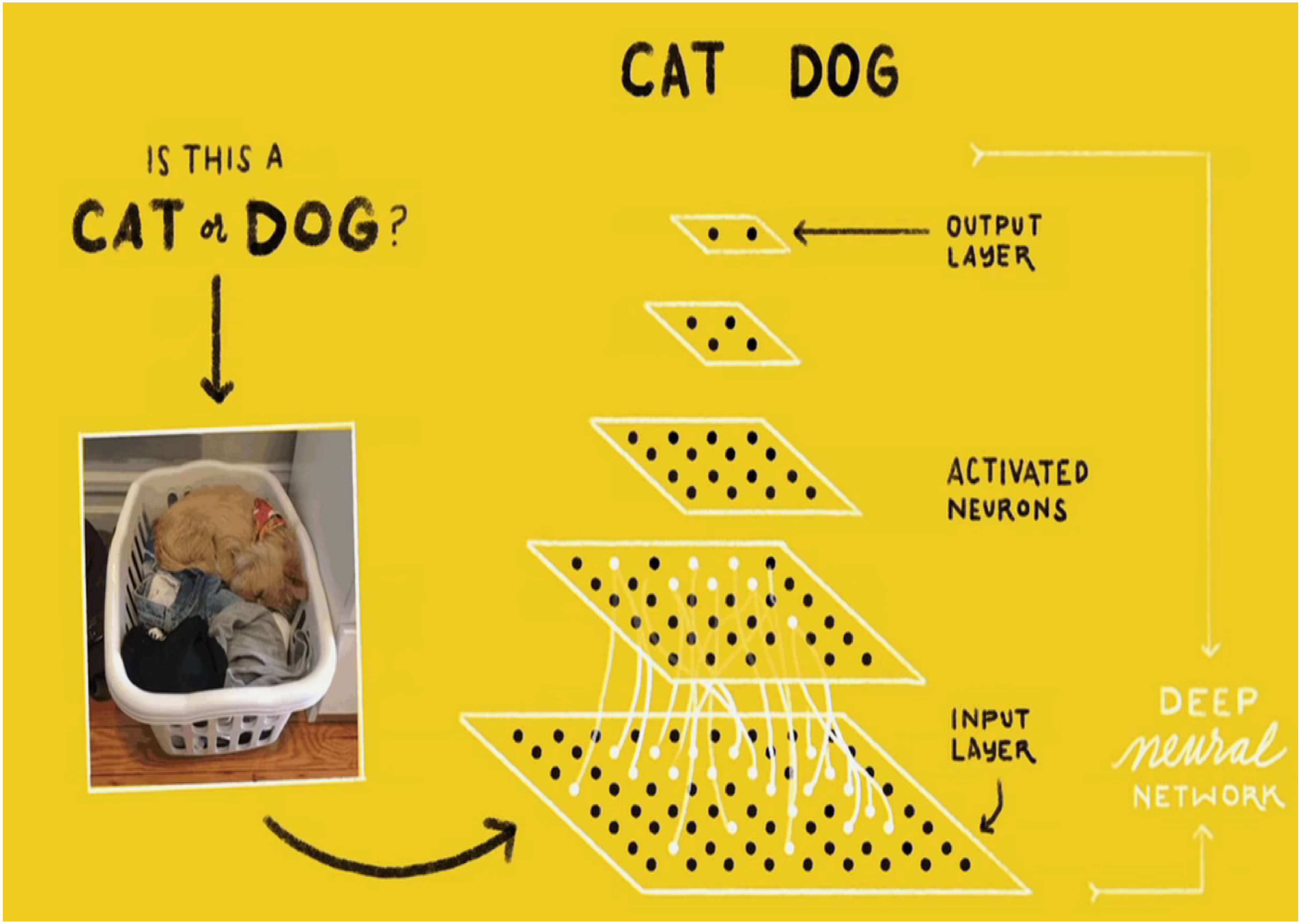
High Dimensional Classification

CalTech 101

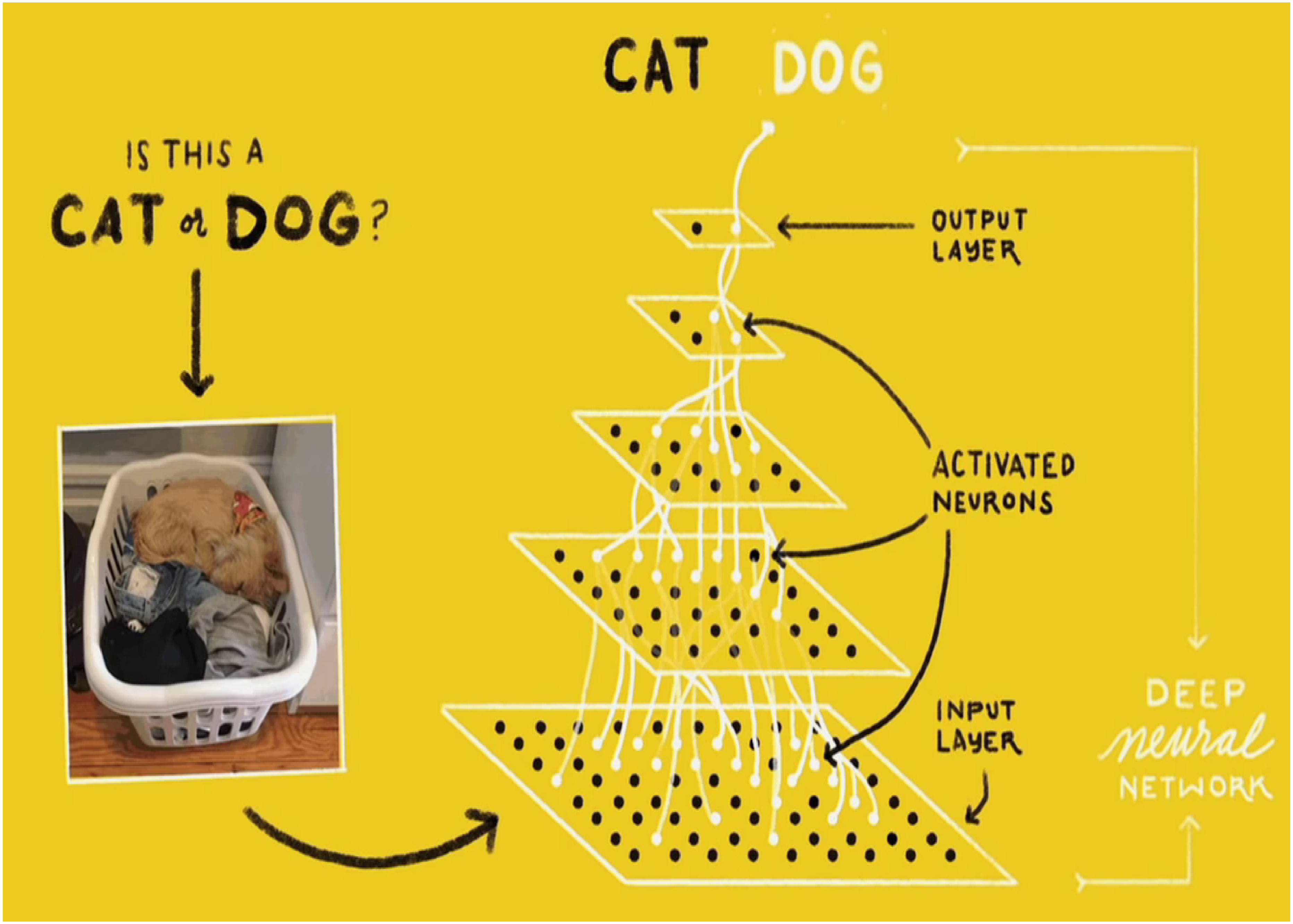
Anchor | Joshua Tree | Beaver | Lotus | Water Lily



- Considerable variability in each class.
- Euclidean distances are meaningless.
- Need to find Informative Invariants.

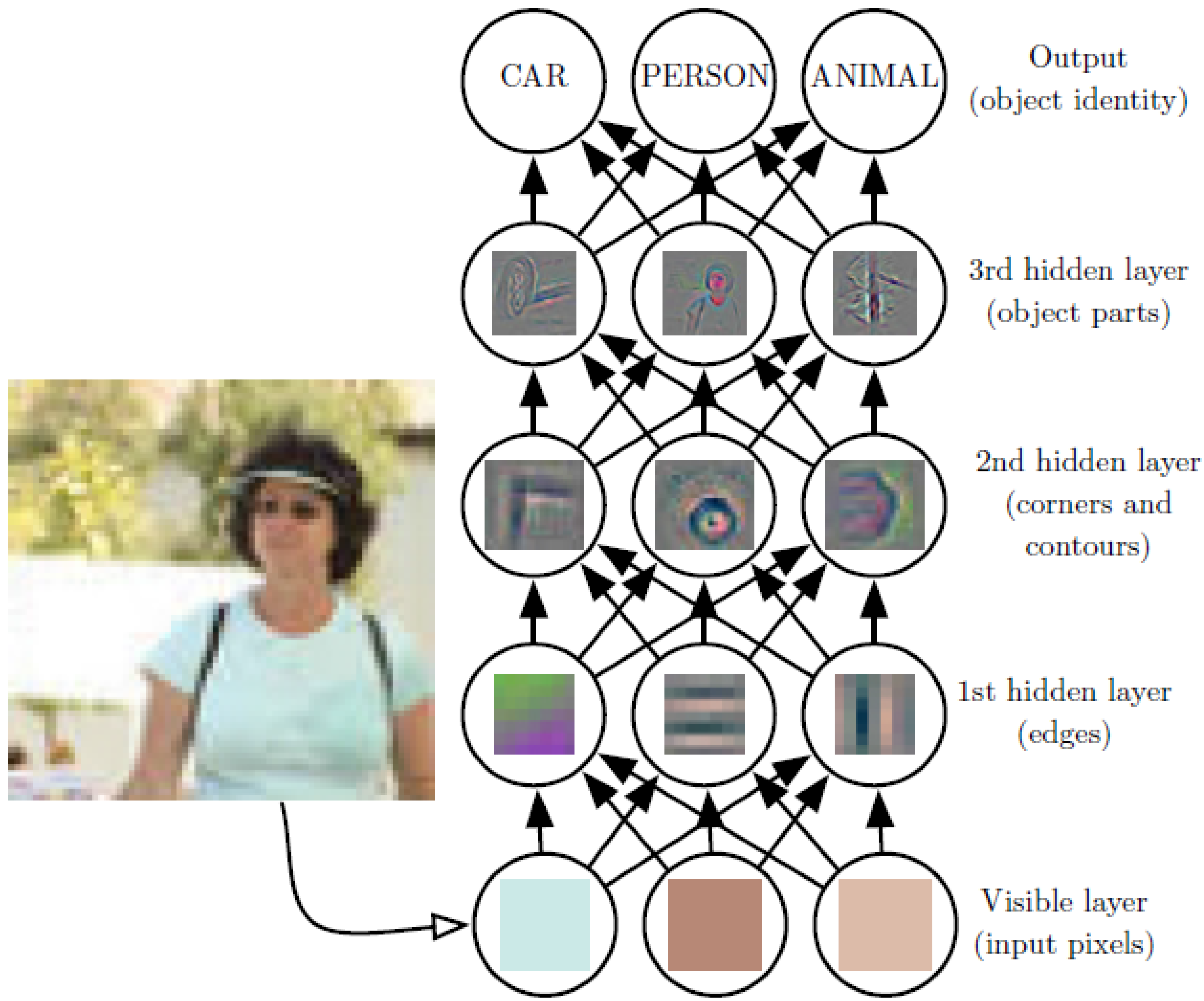


CAT DOG
IS THIS A
CAT or DOG?
OUTPUT LAYER
ACTIVATED NEURONS
INPUT LAYER
DEEP neural NETWORK

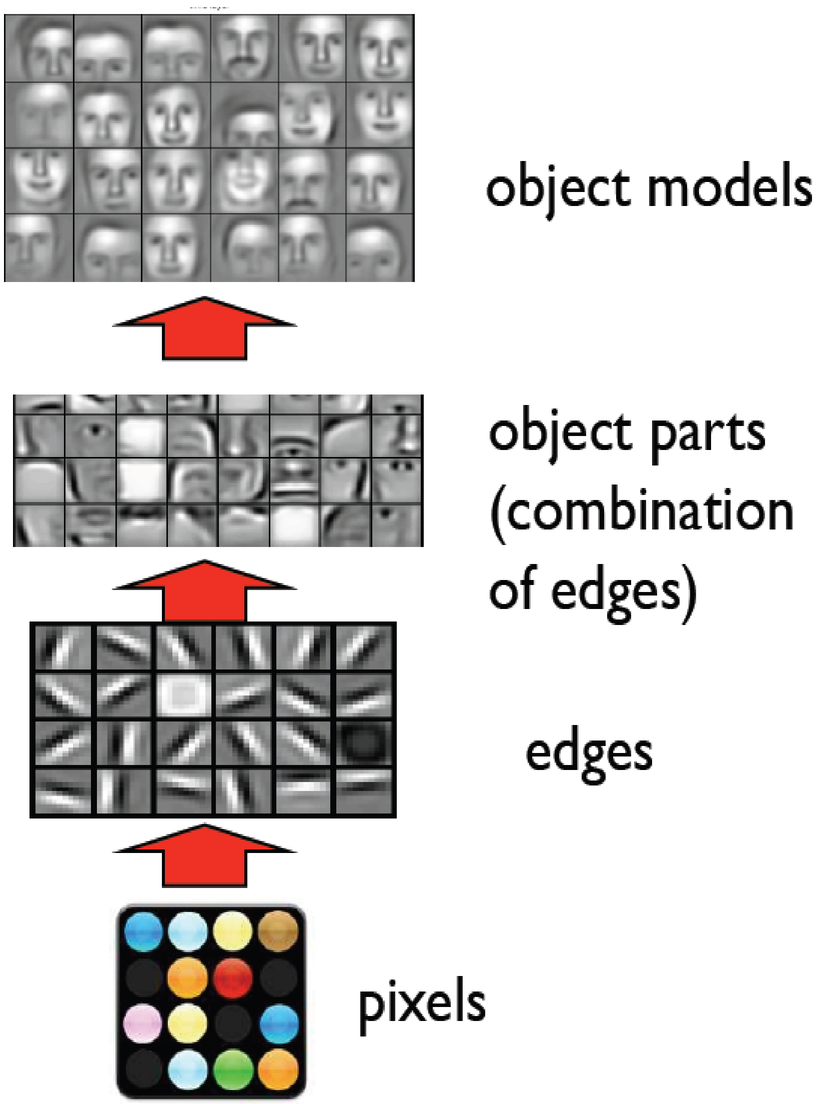


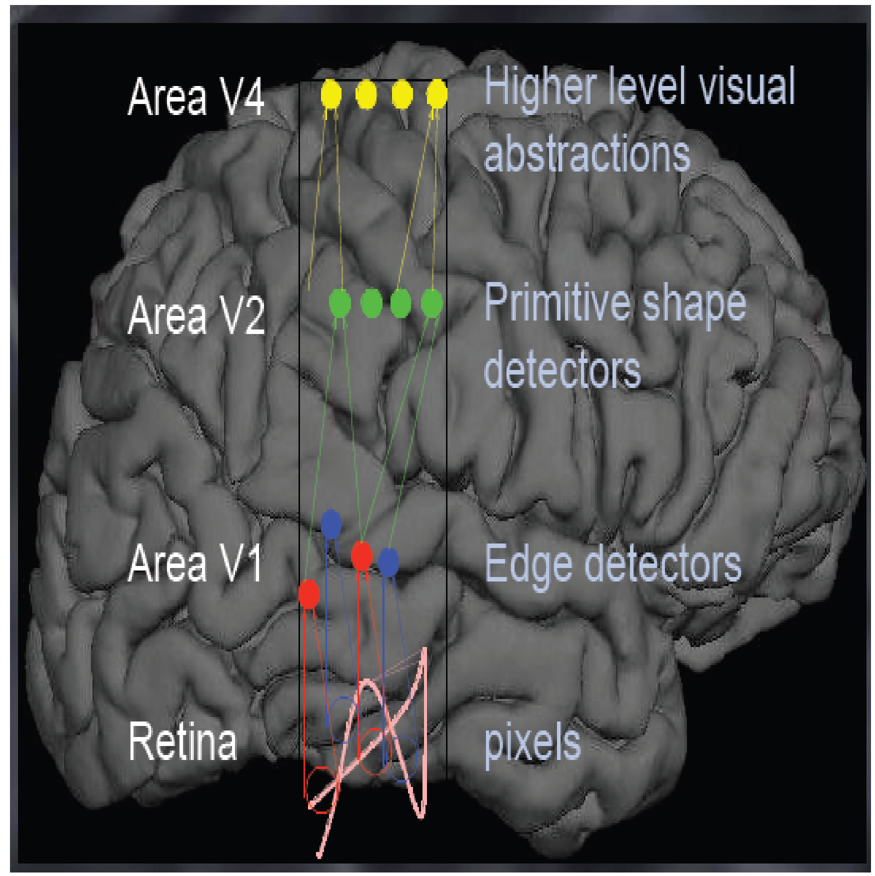
IS THIS A
CAT or DOG?
CAT DOG
OUTPUT LAYER
ACTIVATED NEURONS
INPUT LAYER
DEEP neural NETWORK

Depth: Repeated Composition



Deep learning vs. the brain



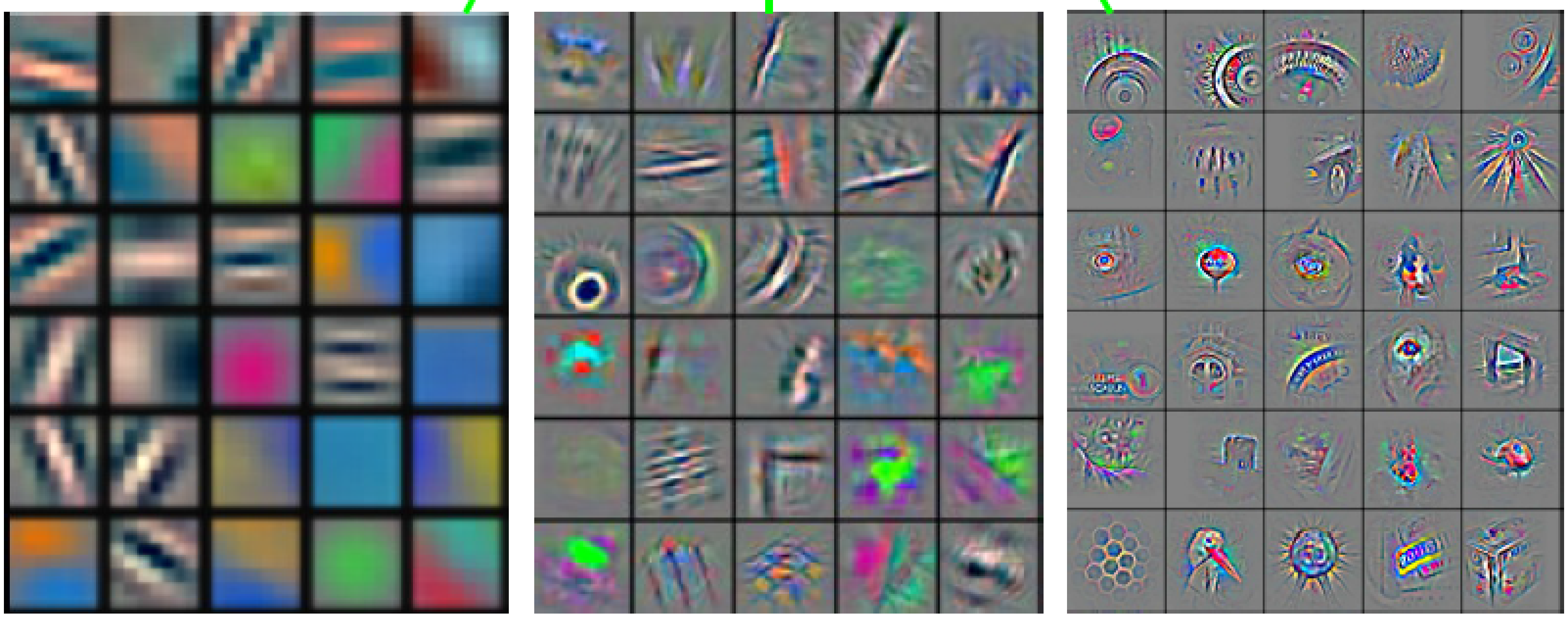


Deep Learning = Learning Hierarchical Representations

Y LeCun
MA Ranzato

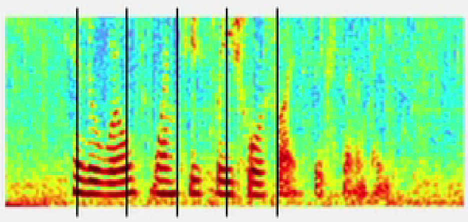
- It's deep if it has more than one stage of non-linear feature transformation

Low-Level Feature → Mid-Level Feature → High-Level Feature → Trainable Classifier →



Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

functions a deep neural network can learn

input		output
Pixels:		"lion"
Audio:		"How cold is it outside?"
"Hello, how are you?"		"Bonjour, comment allez-vous?"
Pixels:		"A blue and yellow train travelling down the tracks"

TensorFlow

http://tensorflow.org/

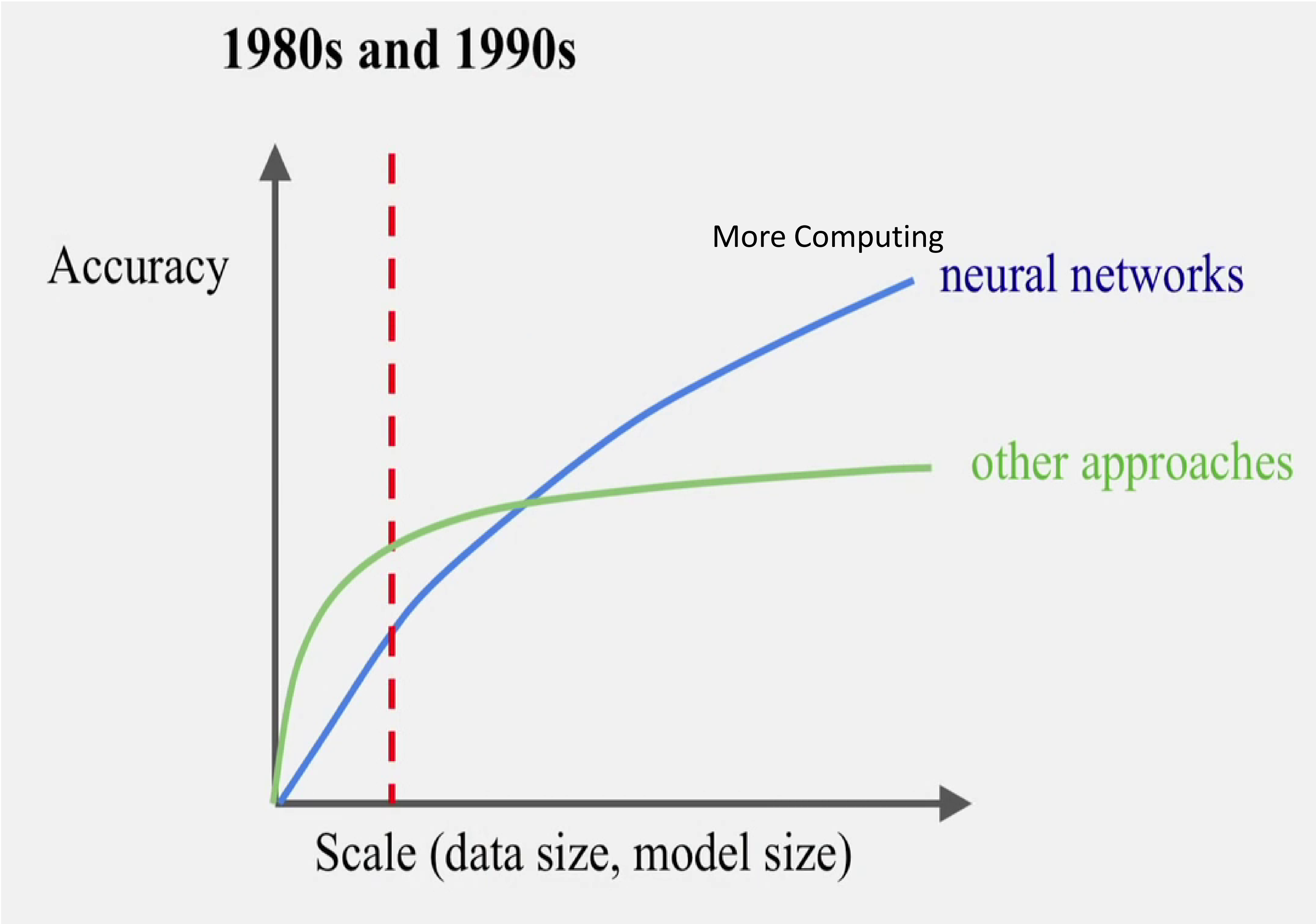
open, standard software for machine learning

great for Deep Learning in particular

1M+
downloads

500+
contributors

#17
most popular GitHub repository


1980s and 1990s
Accuracy
More Computing
neural networks
other approaches
Scale (data size, model size)

more computational power needed

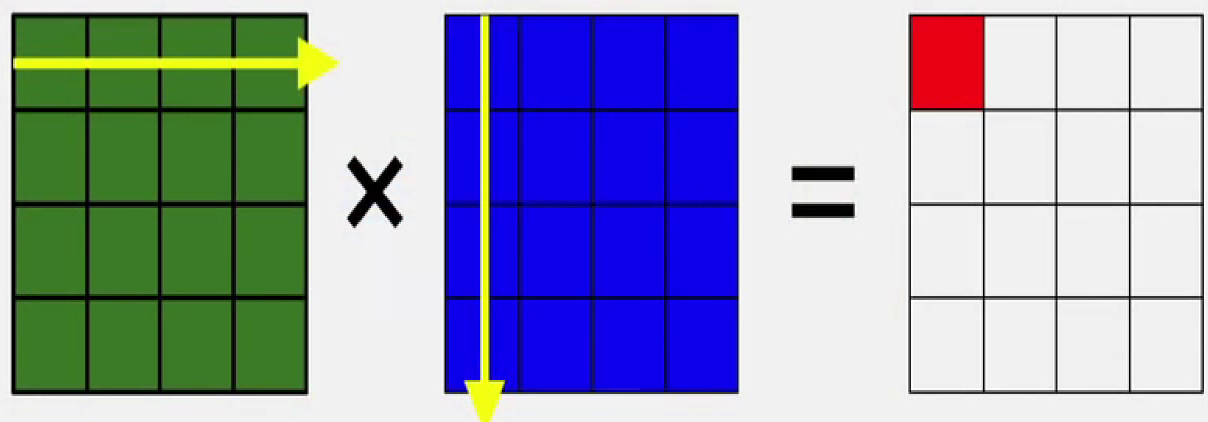
deep learning is transforming how we design computers

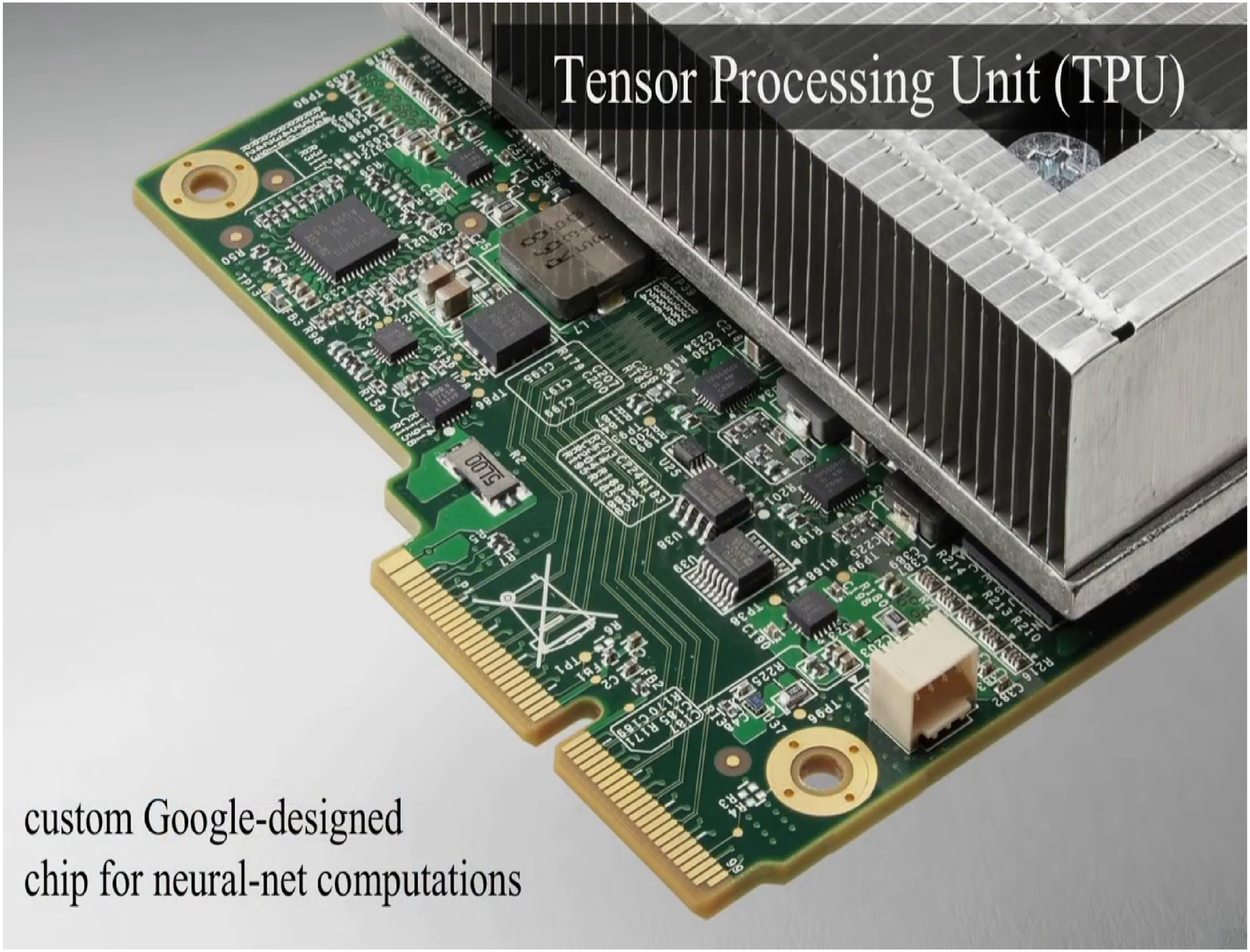
special computation properties

reduced precision ok

$$\begin{array}{r} \text{about } 1.2 \\ \times \text{ about } 0.6 \\ \hline \text{about } 0.7 \end{array} \quad \textbf{NOT} \quad \begin{array}{r} 1.21042 \\ \times\ 0.61127 \\ \hline 0.73989343 \end{array}$$

handful of specific operations




Tensor Processing Unit (TPU)
custom Google-designed
chip for neural-net computations

example queries of the future

Which of these eye images shows symptoms of diabetic retinopathy?

Describe this video in Spanish

Please fetch me a cup of tea from the kitchen

Find me documents related to reinforcement learning for robotics and summarize them in German

AI can be Terrifying


INDEPENDENT News Voices Culture Lifestyle Tech Sport Olympics Daily Edition

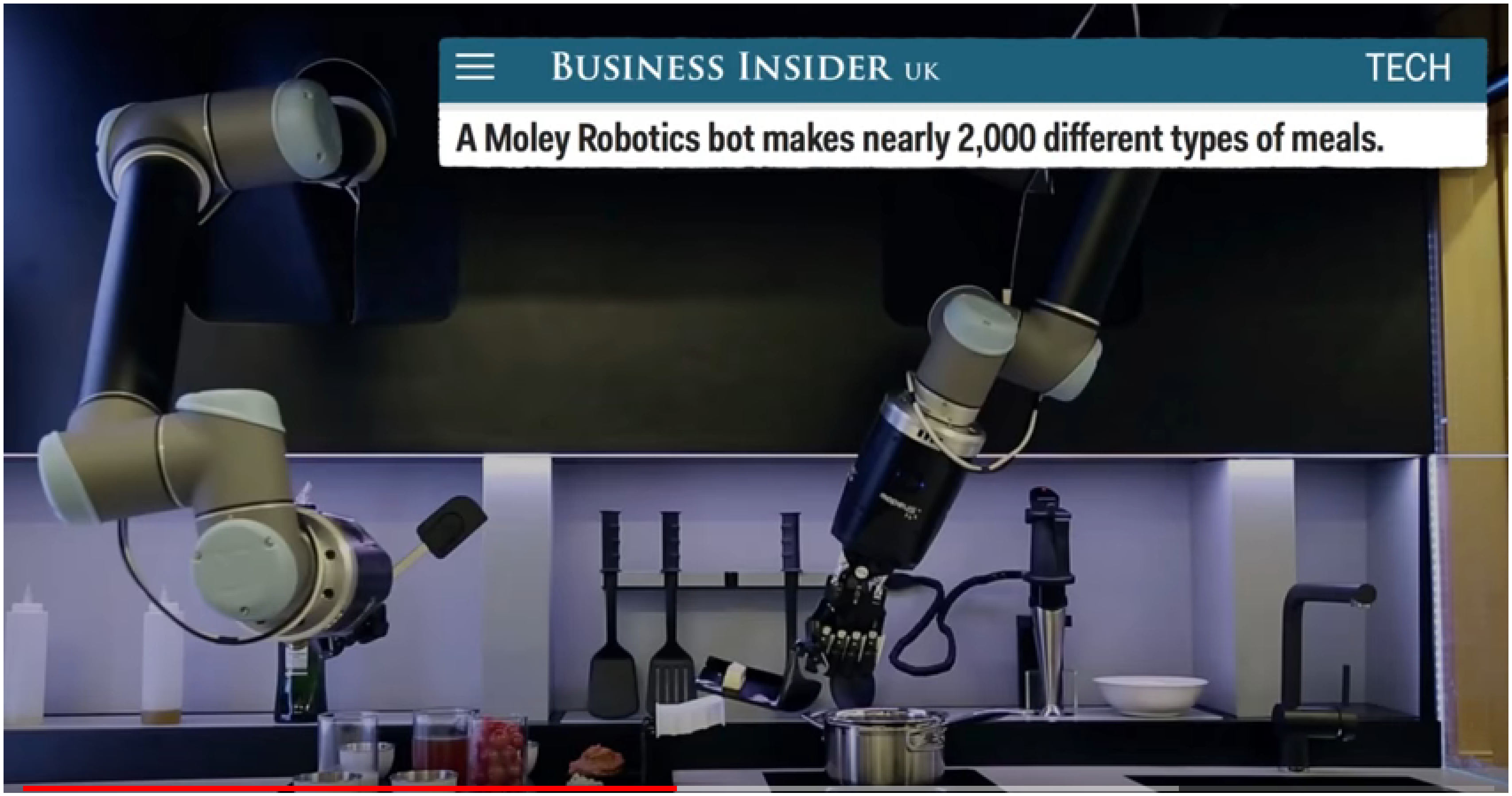
Lifestyle › Tech › News

Apple supplier Foxconn replaces 60,000 workers with robots at China factory

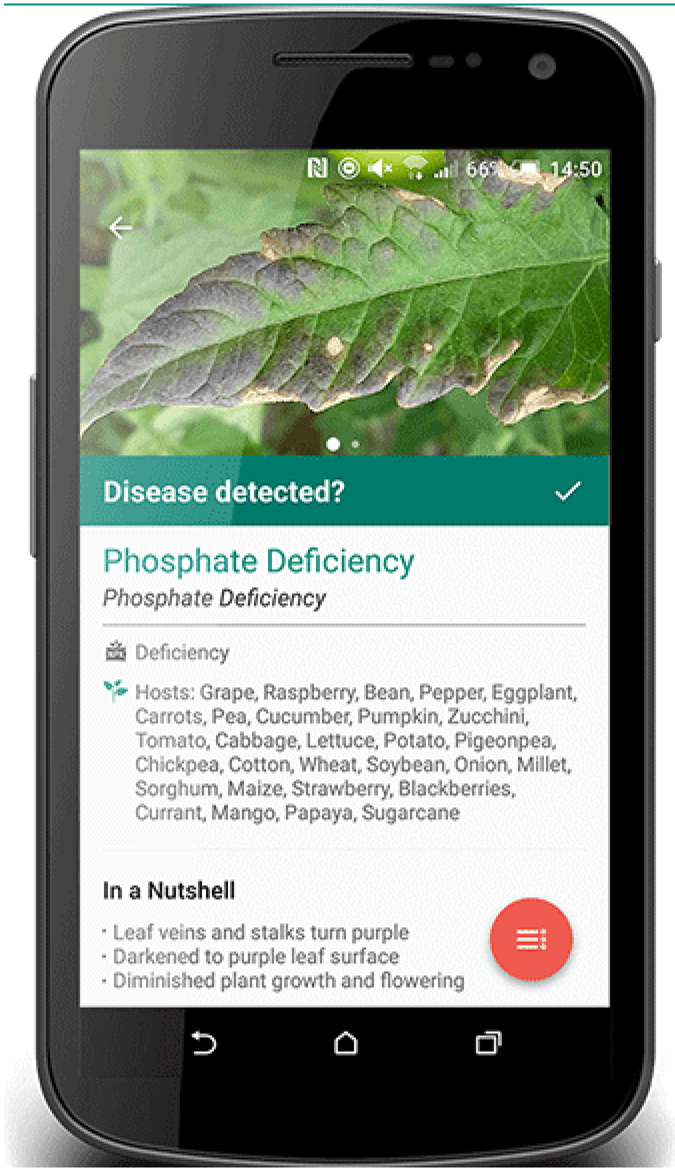
Employee numbers at the Foxconn factory in Kunshan were slashed from 110,000 to 50,000

Doug Bolton | @DougieBolton | Wednesday 25 May 2016 | 1 comment

AI can be Exciting


BUSINESS INSIDER UK
TECH
A Moley Robotics bot makes nearly 2,000 different types of meals.

https://plantix.net/



Instant diagnostics and solutions based on your picture

Image recognition

Plantix analyzes your picture within a few seconds and gives you instant feedback on your plant problem.

Customized management options

In addition to detection results, Plantix gives you a detailed description of possible control solutions - both biological and conventional.

Preventive measures

Plantix offers information on preventive measures to protect your crop from the next attack.

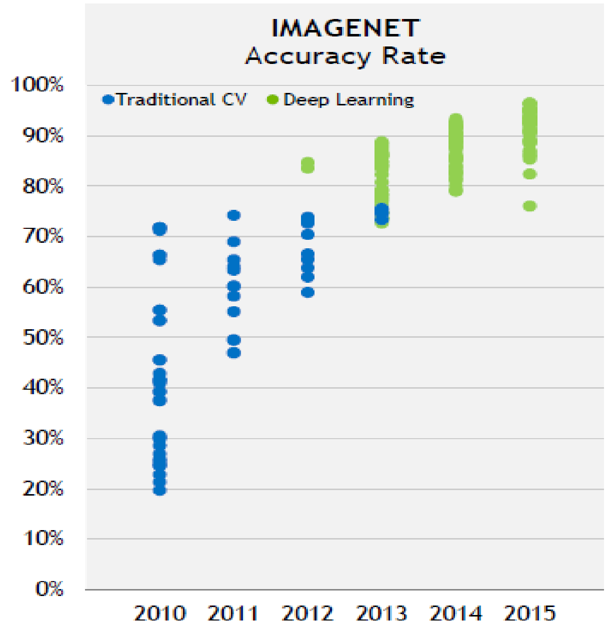
Here is a collection of datasets with images of leaves https://www.plantvillage.org/
http://www.apsnet.org/publications/imageresources/ImageDatabase/Pages/default.aspx
and more generic image datasets that include plant leaves
http://visualgenome.org/
http://image-net.org/
http://www.plant-phenotyping.org/datasets-download http://www.ipmimages.org/
https://archive.ics.uci.edu/ml/datasets/One-hundred+plant+species+leaves+data+set

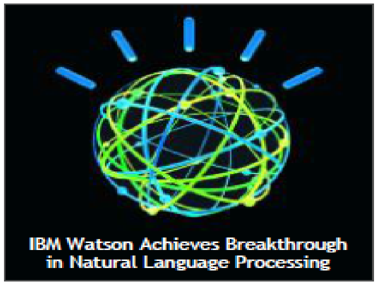
One Accelerating factor of Deep learning in Computer Vision

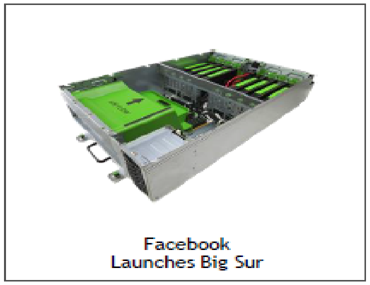
Labeled dataset and competitions

ImageNet Dataset (2011): The Largest Hand-Labeled Dataset in the World

THE AI RACE IS ON







The real breakthrough took place in 2012, when the accuracy rate in the [ImageNet competition] jumped dramatically through the application of deep learning and the abandonment of hand coding.

This past year, Microsoft and Google announced that their neural networks are now beating human capabilities.
Baidu, the Chinese search-giant, announced that it can now beat humans in voice recognition.
Microsoft, working with Hong Kong University of Science and Technology, beat a college student in an IQ test.

Deep Learning
in Fluid Dynamics ?.

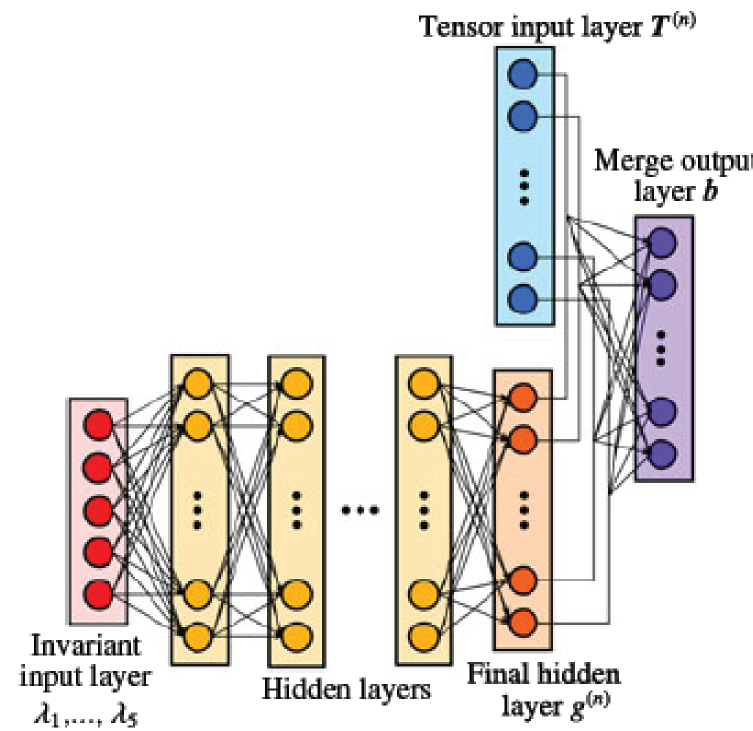
What the genie can do here?.

Focus on **Fluids** journals.cambridge.org/focus

Deep learning in fluid dynamics

J. Nathan Kutz†

Department of Applied Mathematics, University of Washington, Seattle, WA 98195, USA



Given the computational maturity of DNNs and how readily available they are (see Google's open source software called TensorFlow: tensorflow.org), it is perhaps time for part of the turbulence modelling community to adopt what has become an important and highly successful part of the machine learning culture: challenge data sets. Donoho argues (Donoho 2015), and I am in complete agreement, that challenge data sets allow researchers a fair comparison of their DNN innovations on training data (publicly available to all) and test data (not publicly available, but accessible with your algorithm). Importantly, this would give the fluids community their own ImageNet data sets to help generate reproducible and validated performance gains on DNNs for applications on complex flows. Perhaps Ling, Kurzawski and Templeton can help push the community forward in this way.

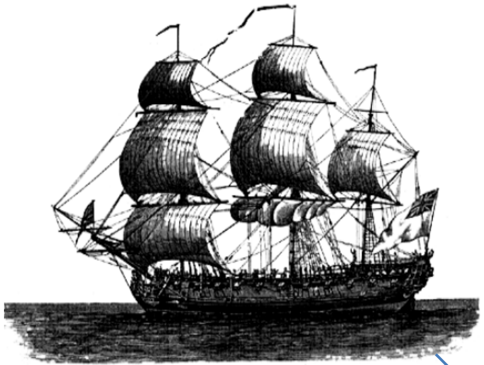
How do we learn?

We learn from examples and repeated practice

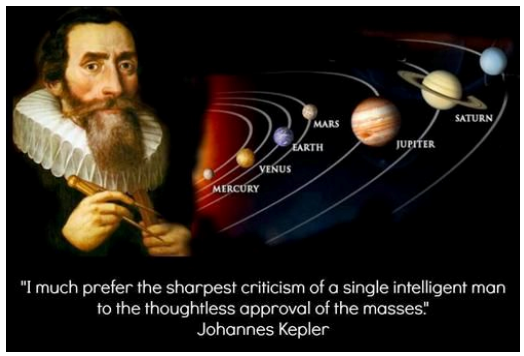
In ML, Computers are given examples (data) and is asked to learn (to classify, identify relationships etc)

Can we apply machine learning to get equations of dynamics from data?.

Kepler to Newton



Thousands of years of data on stars' and planets' positions



Johannes Kepler: 1571-1630:

First Data Scientist

50-60 years

Derived equations of dynamics that fitted the data

Newton: 1643-1727

PDE-NET: LEARNING PDES FROM DATA

Anonymous authors
Paper under double-blind review

Under review as a conference paper at ICLR 2018

International Conference on Learning Representations

Overview

The performance of machine learning methods is heavily dependent on the choice of data representation (or features) on which they are applied. The rapidly developing field of deep learning is concerned with questions surrounding how we can best learn meaningful and useful representations of data

Abstract

Partial differential equations (PDEs) play a prominent role in many disciplines such as applied mathematics, physics, chemistry, material science, computer science, etc. PDEs are commonly derived based on physical laws or empirical observations. However, the governing equations for many complex systems in modern applications are still not fully known. With the rapid development of sensors, computational power, and data storage in the past decade, huge quantities of data can be easily collected and efficiently stored. Such vast quantity of data offers new opportunities for data-driven discovery of hidden physical laws. Inspired by the latest development of neural network designs in deep learning, we propose a new feed-forward deep network, called PDE-Net, to fulfill two objectives at the same time: to accurately predict dynamics of complex systems and to uncover the underlying hidden PDE models. The basic idea of the proposed PDE-Net is to learn differential operators by learning convolution kernels (filters), and apply neural networks or other machine learning methods to approximate the unknown nonlinear responses. Comparing with existing approaches, which either assume the form of the nonlinear response is known or fix certain finite difference approximations of differential operators, our approach has the most flexibility by learning both differential operators and the nonlinear responses. A special feature of the proposed PDE-Net is that all filters are properly constrained, which enables us to easily identify the governing PDE models while still maintaining the expressive and predictive power of the network. These constrains are carefully designed by fully exploiting the relation between the orders of differential operators and the orders of sum rules of filters (an important concept originated from wavelet theory). We also discuss relations of the PDE-Net with some existing networks in computer vision such as Network-In-Network (NIN) and Residual Neural Network (ResNet). Numerical experiments show that the PDE-Net has the potential to uncover the hidden PDE of the observed dynamics, and predict the dynamical behavior for a relatively long time, even in a noisy environment.

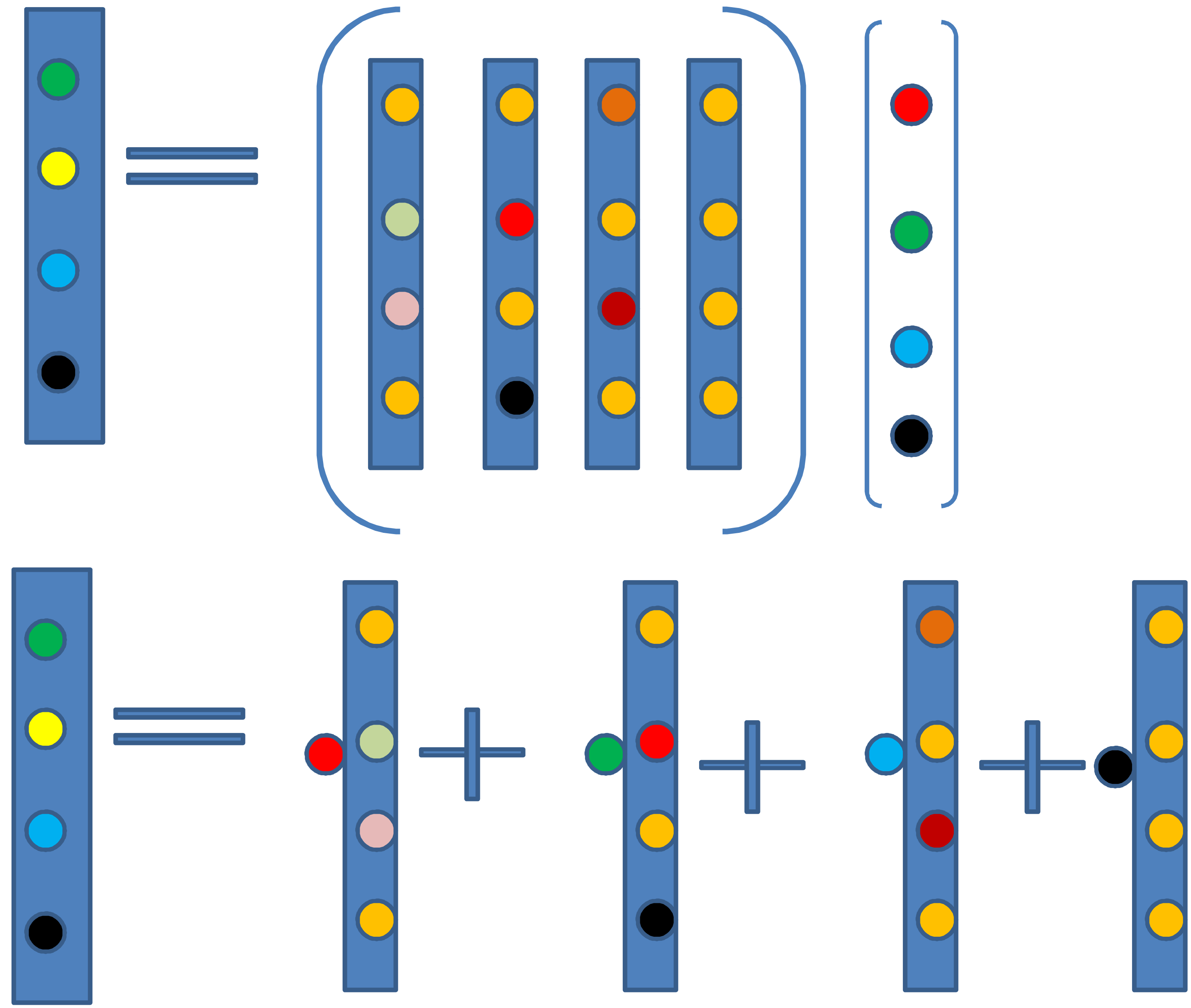
DATA-DRIVEN DISCOVERY OF DYNAMICAL SYSTEMS IN THE ENGINEERING, PHYSICAL AND BIOLOGICAL SCIENCES

J. Nathan Kutz

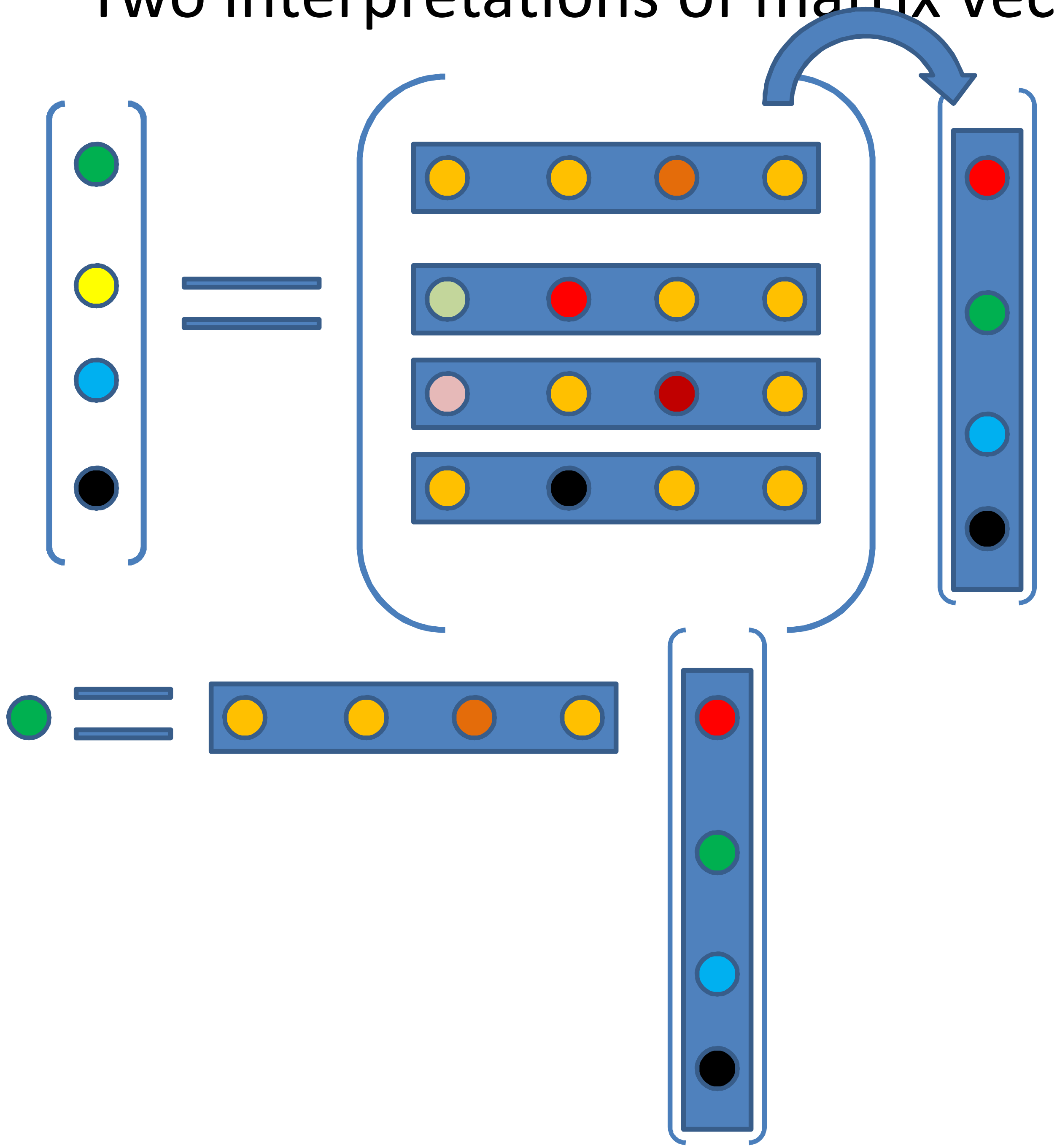
Department of Applied Mathematics
University of Washington
Seattle, WA 98195-3925
Email: kutz@uw.edu

$$Ax=b$$

Two interpretations of matrix vector product



Two interpretations of matrix vector product

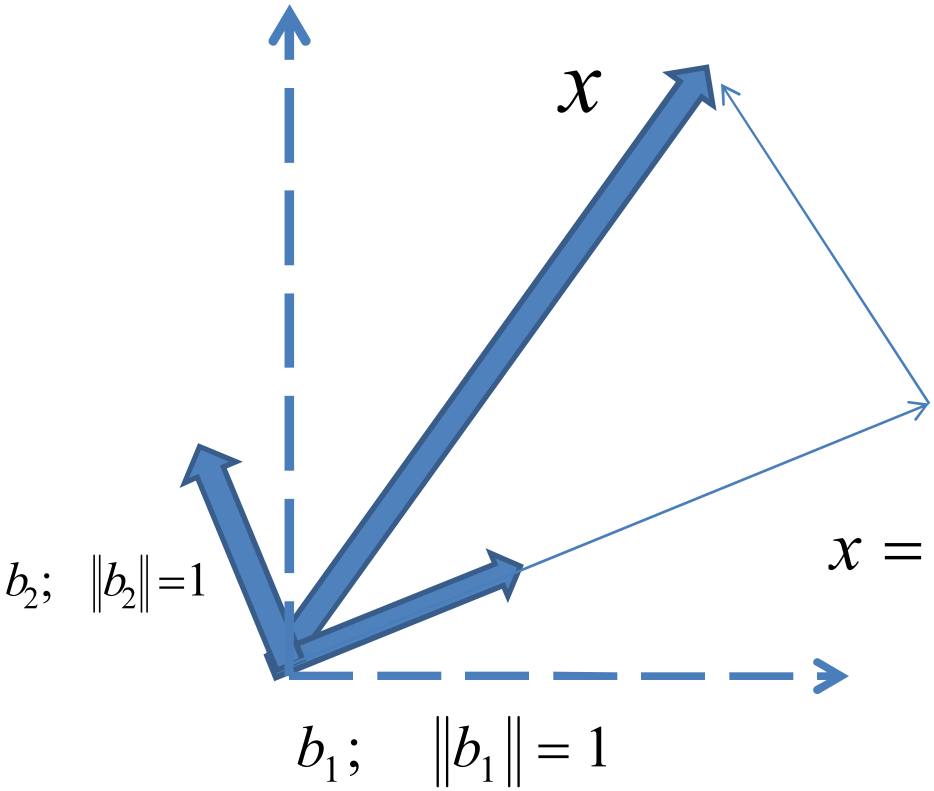


Force Resolution

$$x \equiv \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

$$b_1 \equiv \begin{pmatrix} b_{11} \\ b_{12} \end{pmatrix}$$

$$b_1 \equiv \begin{pmatrix} b_{21} \\ b_{22} \end{pmatrix}$$



$$x = \left(x^T b_1\right) b_1 + \left(x^T b_2\right) b_2$$

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_{11} & b_{21} \\ b_{12} & b_{22} \end{pmatrix} \begin{pmatrix} x^T b_1 \\ x^T b_2 \end{pmatrix}$$

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \left(x^T b_1\right) \begin{pmatrix} b_{11} \\ b_{12} \end{pmatrix} + \left(x^T b_2\right) \begin{pmatrix} b_{21} \\ b_{22} \end{pmatrix}$$

Point in 4-Dimensional space

$$\boldsymbol{x} = \left(\boldsymbol{x}^T \boldsymbol{Base_1}\right)\boldsymbol{Base_1} + \left(\boldsymbol{x}^T \boldsymbol{Base_2}\right)\boldsymbol{Base_2} + \left(\boldsymbol{x}^T \boldsymbol{Base_3}\right)\boldsymbol{Base_3} + \left(\boldsymbol{x}^T \boldsymbol{Base_4}\right)\boldsymbol{Base_4}$$

$$\boldsymbol{x} = (base_1^T \boldsymbol{x})\, Base_1 + (base_2^T \boldsymbol{x})\, Base_2 + (base_3^T \boldsymbol{x})\, Base_3 + (base_4^T \boldsymbol{x})\, Base_4$$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = X_1\, Base_1 + X_2\, Base_2 + X_3\, Base_3 + X_4\, Base_4 = \begin{bmatrix} | & | & | & | \\ Base_1 & Base_2 & Base_3 & Base_4 \\ | & | & | & | \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix}$$

$$\begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} = \begin{bmatrix} base_1^T \boldsymbol{x} \\ base_2^T \boldsymbol{x} \\ base_3^T \boldsymbol{x} \\ base_4^T \boldsymbol{x} \end{bmatrix}$$

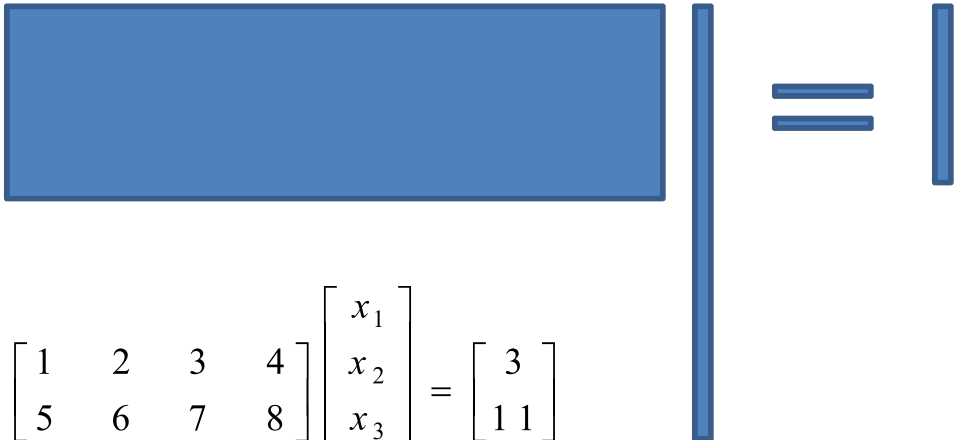
$$\underbrace{\begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} = \begin{bmatrix} \leftarrow base_1^T \rightarrow \\ \leftarrow base_2^T \rightarrow \\ \leftarrow base_3^T \rightarrow \\ \leftarrow base_4^T \rightarrow \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}}_{\textit{Forward}\ \text{Transform}}$$

$$\mathrm{X} = \mathrm{B}^T x$$

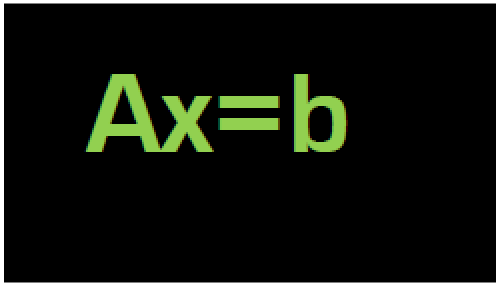
$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \underbrace{\begin{bmatrix} | & | & | & | \\ Base_1 & Base_2 & Base_3 & Base_4 \\ | & | & | & | \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix}}_{\textit{inverse}\ \text{Transform}}$$

$$x = \mathrm{BX}$$

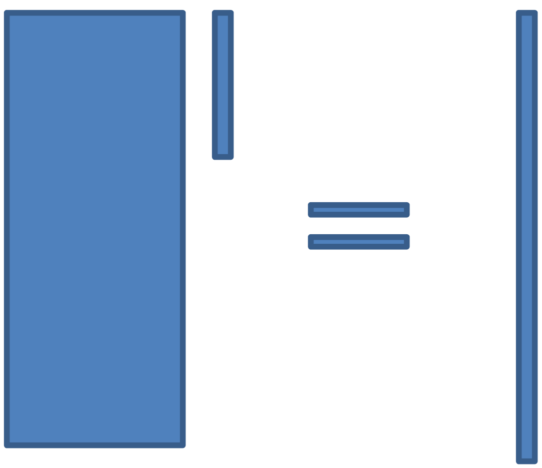
Under-determined System

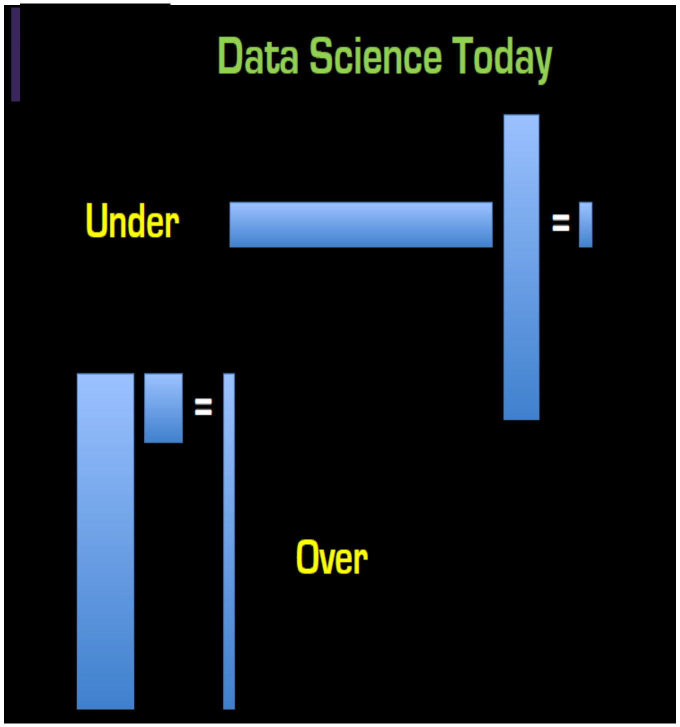


$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 3 \\ 11 \end{bmatrix}$$



Over-determined System





Sparse solution

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 3 \\ 11 \end{bmatrix} \qquad A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix} \qquad b = \begin{bmatrix} 3 \\ 11 \end{bmatrix}$$

$$Ax = b$$

In matlab

$$x = A \backslash b$$

$$x = pinv(A) * b$$

[redacted]

```
>> pinv(A)*b

ans =

    1.1000
    0.7000
    0.3000
   -0.1000
```

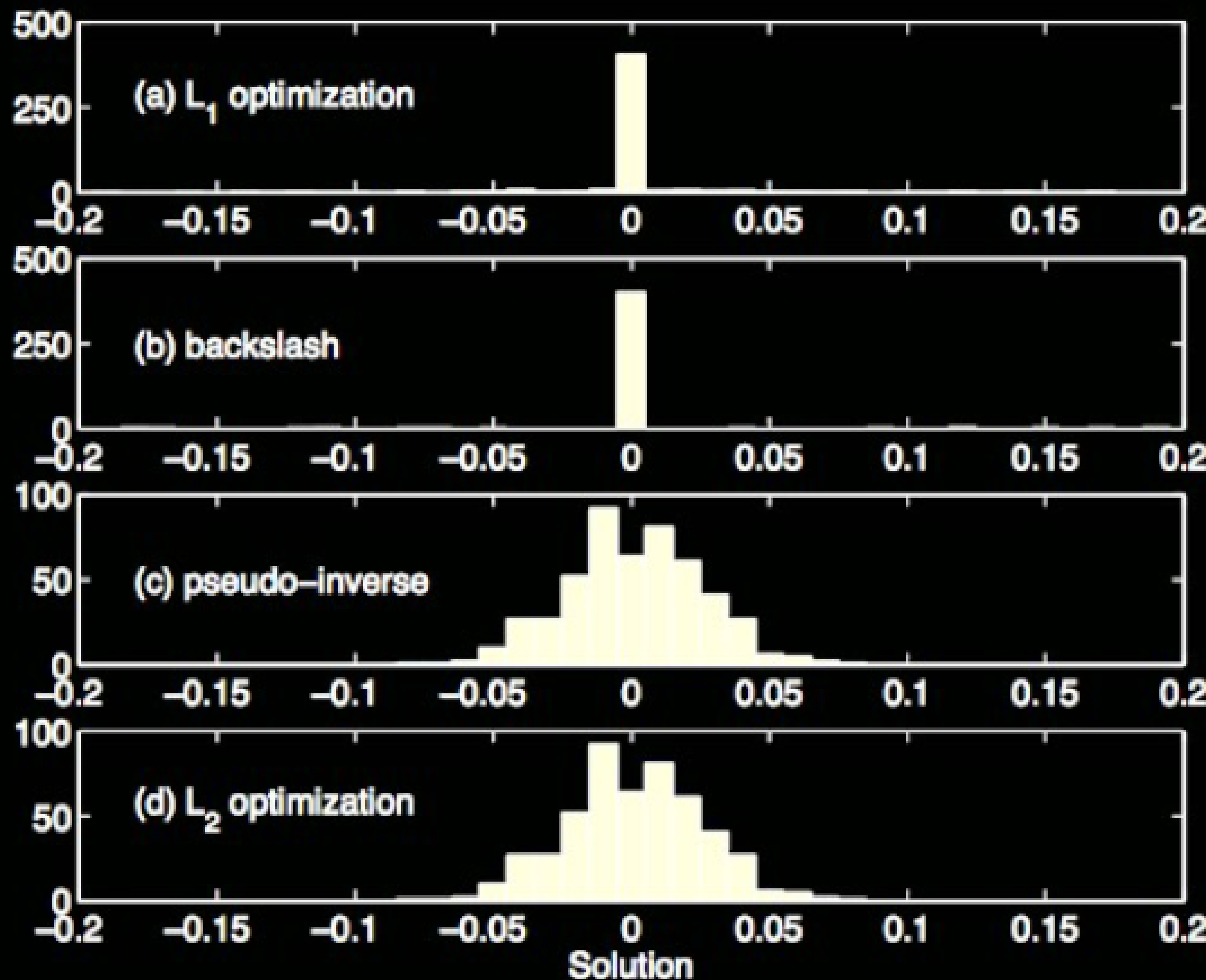
$$\min_x \|x\|_1 \quad st \quad \mathrm{Ax=b}$$

$$\min_x \|x\|_2 \quad st \quad \mathrm{Ax=b}$$

Solving Ax=b

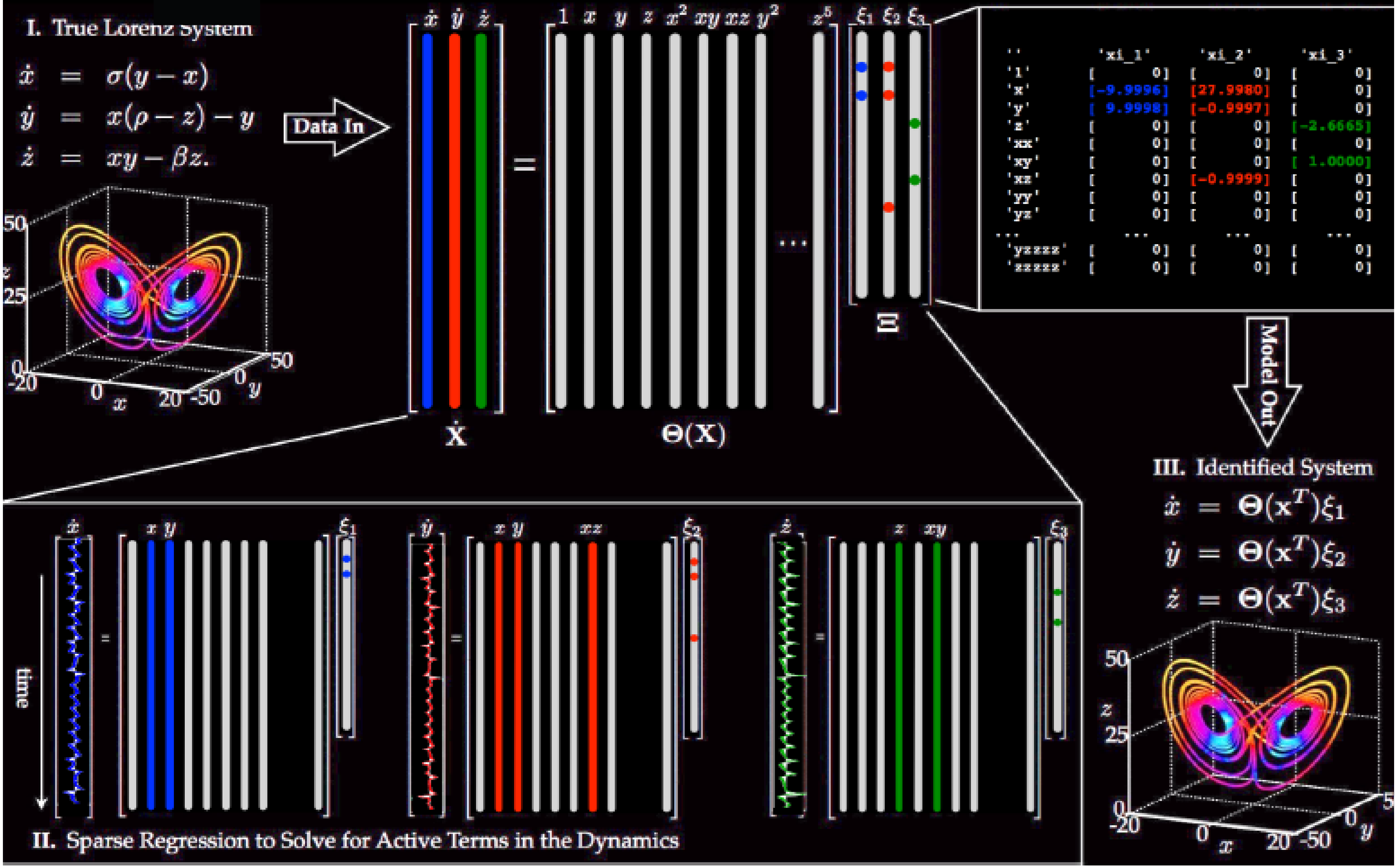
```
m=100; n=500;
A=randn(m,n);
b=randn(m,1);

x1=A\b;
x2=pinv(A)*b;
```



Lasso & Ridge & others

Sparse Identification of Nonlinear Dynamics (SINDy)



Nonlinear Systems ID

1. Collect Data

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}^T(t_1) \\ \mathbf{x}^T(t_2) \\ \vdots \\ \mathbf{x}^T(t_m) \end{bmatrix} = \overbrace{\begin{bmatrix} x_1(t_1) & x_2(t_1) & \cdots & x_n(t_1) \\ x_1(t_2) & x_2(t_2) & \cdots & x_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(t_m) & x_2(t_m) & \cdots & x_n(t_m) \end{bmatrix}}^{\text{state}} \Big\downarrow \text{time}$$

$$\dot{\mathbf{X}} = \begin{bmatrix} \dot{\mathbf{x}}^T(t_1) \\ \dot{\mathbf{x}}^T(t_2) \\ \vdots \\ \dot{\mathbf{x}}^T(t_m) \end{bmatrix} = \begin{bmatrix} \dot{x}_1(t_1) & \dot{x}_2(t_1) & \cdots & \dot{x}_n(t_1) \\ \dot{x}_1(t_2) & \dot{x}_2(t_2) & \cdots & \dot{x}_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \dot{x}_1(t_m) & \dot{x}_2(t_m) & \cdots & \dot{x}_n(t_m) \end{bmatrix}.$$

2. Build Library of Candidate Nonlinearities

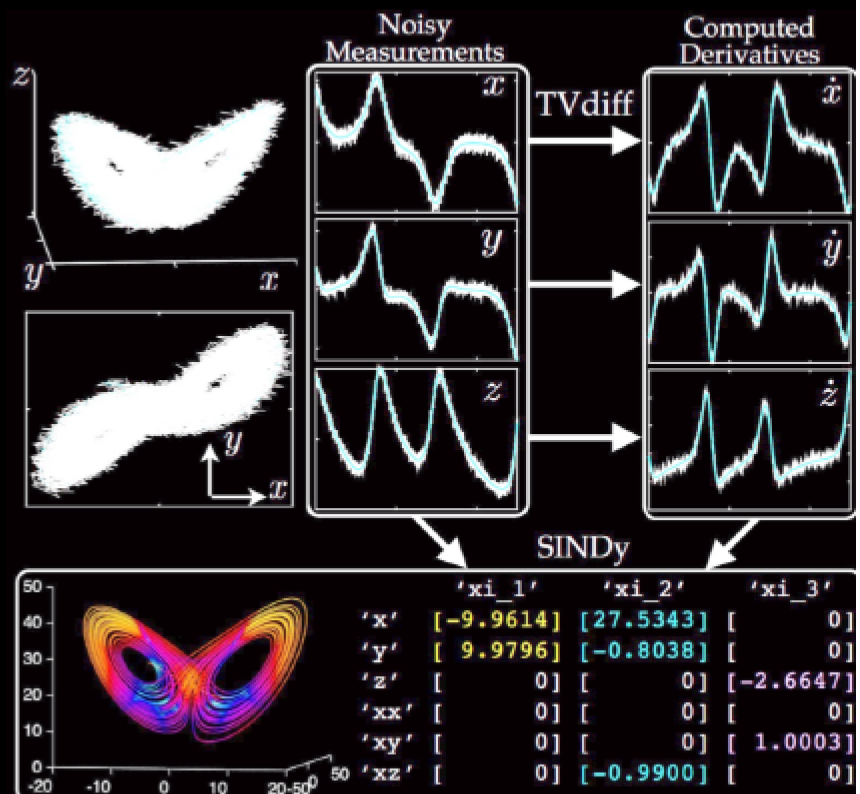
$$\Theta(\mathbf{X}) = \begin{bmatrix} | & | & | & | & & | & | & \\ 1 & \mathbf{X} & \mathbf{X}^{P_2} & \mathbf{X}^{P_3} & \cdots & \sin(\mathbf{X}) & \cos(\mathbf{X}) & \cdots \\ | & | & | & | & & | & | & \end{bmatrix}.$$

3. Sparse Regression to Find Active Terms

$$\dot{\mathbf{X}} = \Theta(\mathbf{X})\Xi.$$

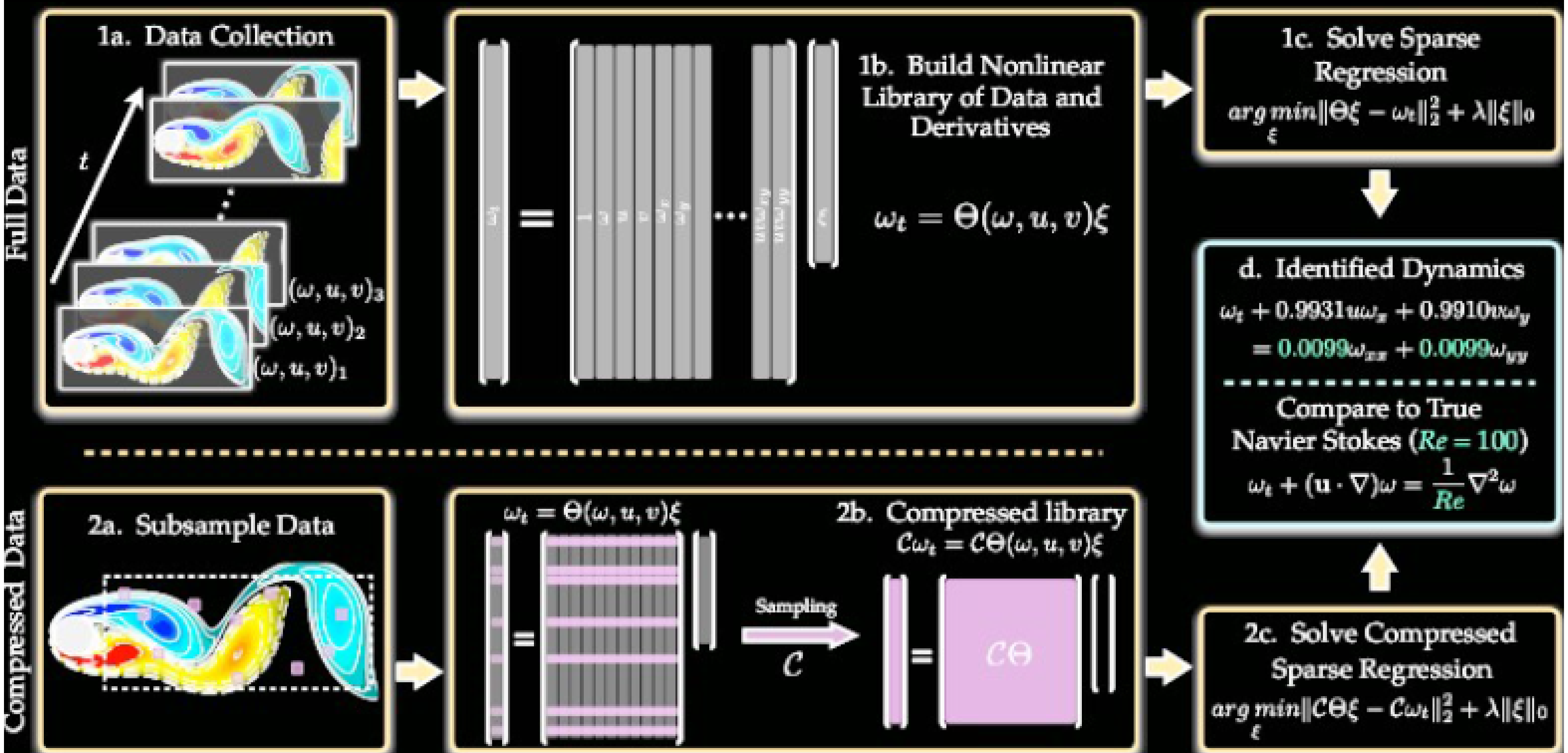
4. Nonlinear Model

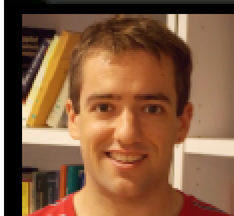
$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) = \Xi^T(\Theta(\mathbf{x}^T))^T$$



Total Variation Regularized Numerical Differentiation (TVDiff)

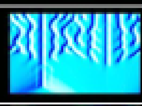
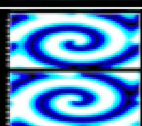
Discovering PDEs





Sam Rudy, Brunton, Proctor & Kutz, Sci Adv (2017)

See also Schaeffer, Proc Roy Soc A (2017)

PDE	Form	Error (no noise, noise)	Discretization
 KdV	$u_t + 6uu_x + u_{xxx} = 0$	$1\%\pm0.2\%, 7\%\pm5\%$	$x\in[-30, 30], n{=}512,\ t\in[0, 20], m{=}201$
 Burgers	$u_t + uu_x - \epsilon u_{xx} = 0$	$0.15\%\pm0.06\%, 0.8\%\pm0.6\%$	$x\in[-8, 8], n{=}256,\ t\in[0, 10], m{=}101$
 Schrodinger	$iu_t + \frac{1}{2}u_{xx} - \frac{x^2}{2}u = 0$	$0.25\%\pm0.01\%, 10\%\pm7\%$	$x\in[-7.5, 7.5], n{=}512,\ t\in[0, 10], m{=}401$
 NLS	$iu_t + \frac{1}{2}u_{xx} + \lvert u\rvert^2 u = 0$	$0.05\%\pm0.01\%, 3\%\pm1\%$	$x\in[-5, 5], n{=}512,\ t\in[0, \pi], m{=}501$
 KS	$u_t + uu_x + u_{xx} + u_{xxxx} = 0$	$1.3\%\pm1.3\%, 70\%\pm27\%$	$x\in[0, 100], n{=}1024,\ t\in[0, 100], m{=}251$
 R-D	$u_t = 0.1\nabla^2 u + \lambda(A)u - \omega(A)v$ $v_t = 0.1\nabla^2 v + \omega(A)u + \lambda(A)v$ $A{=}u^2{+}v^2, \omega{=}-\beta A^2, \lambda{=}1-A^2$	$0.02\% \pm 0.01\%,\ 3.8\% \pm 2.4\%$	$x, y\in[-10, 10], n{=}256,\ t\in[0, 10], m{=}201$ subsample $3 \cdot 10^5$
 Navier Stokes	$\omega_t + (\mathbf{u} \cdot \nabla)\omega = \frac{1}{Re}\nabla^2\omega$	$1\% \pm 0.2\%$, $7\% \pm 6\%$	$x\in[0, 9], n_x{=}449,\ y\in[0, 4], n_y{=}199,$ $t\in[0, 30], m{=}151,$ subsample $3 \cdot 10^5$

Accelerating Eulerian Fluid Simulation With Convolutional Networks

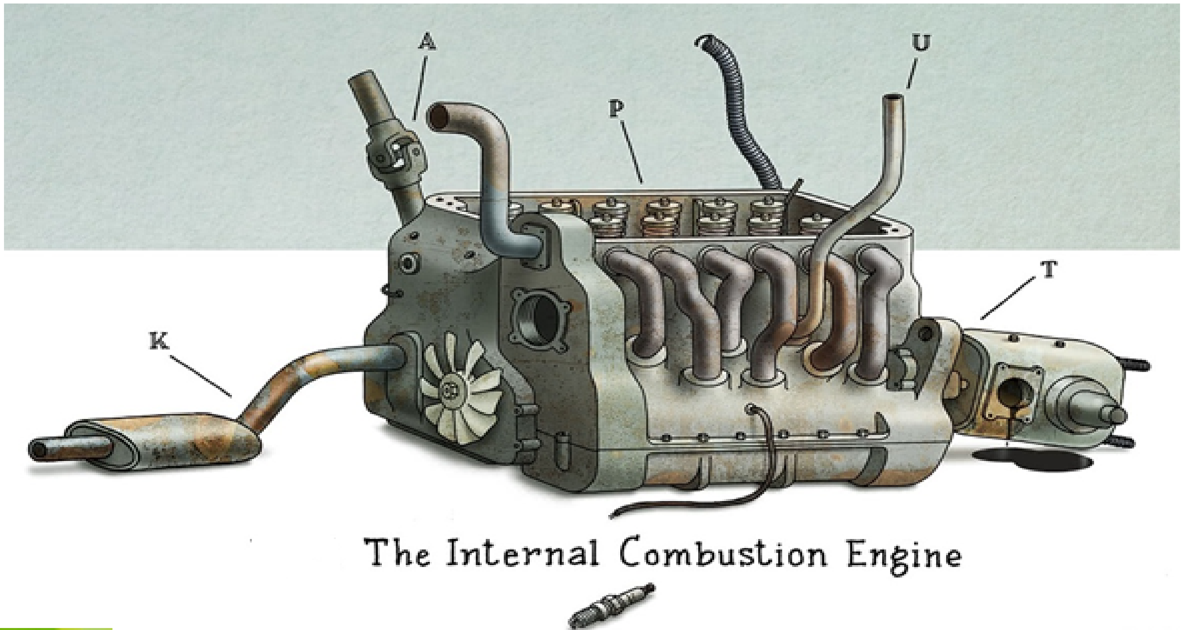
Jonathan Tompson [1] **Kristofer Schlachter** [2] **Pablo Sprechmann** [2 3] **Ken Perlin** [2]

Abstract

Efficient simulation of the Navier-Stokes equations for fluid flow is a long standing problem in applied mathematics, for which state-of-the-art methods require large compute resources. In this work, we propose a data-driven approach that leverages the approximation power of deep-learning with the precision of standard solvers to obtain fast and highly realistic simulations. Our method solves the incompressible Euler equations using the standard operator splitting method, in which a large sparse linear system with many free parameters must be solved. We use a Convolutional Network with a highly tailored architecture, trained using a novel unsupervised learning framework to solve the linear system. We present real-time 2D and 3D simulations that outperform recently proposed data-driven methods; the obtained results are realistic and show good generalization properties.

The dynamics of a large number of physical phenomenon are governed by the incompressible Navier-Stokes equations, which are a set of partial differential equations that must hold throughout a fluid velocity field for all time steps. There are two main computational approaches for simulating these equations: the Lagrangian methods, that approximate continuous quantities using discrete moving particles (Gingold & Monaghan, 1977), and the Eulerian, methods that approximate quantities on a fixed grid (Foster & Metaxas, 1996). We adopt the latter for this work.

Eulerian methods are able to produce accurate results simulating fluids like water with high compute costs. The most demanding portion of this method is the "pressure projection" step, which satisfies an incompressibility constraint. It involves solving the discrete Poisson equation and leads to a well-known sparse, symmetric and positive-definite linear system. Exact solutions can be found via traditional convex optimization techniques, such as the Preconditioned Conjugate Gradient (PCG) algorithm or via stationary iterative methods, like the Jacobi or Gauss-Seidel



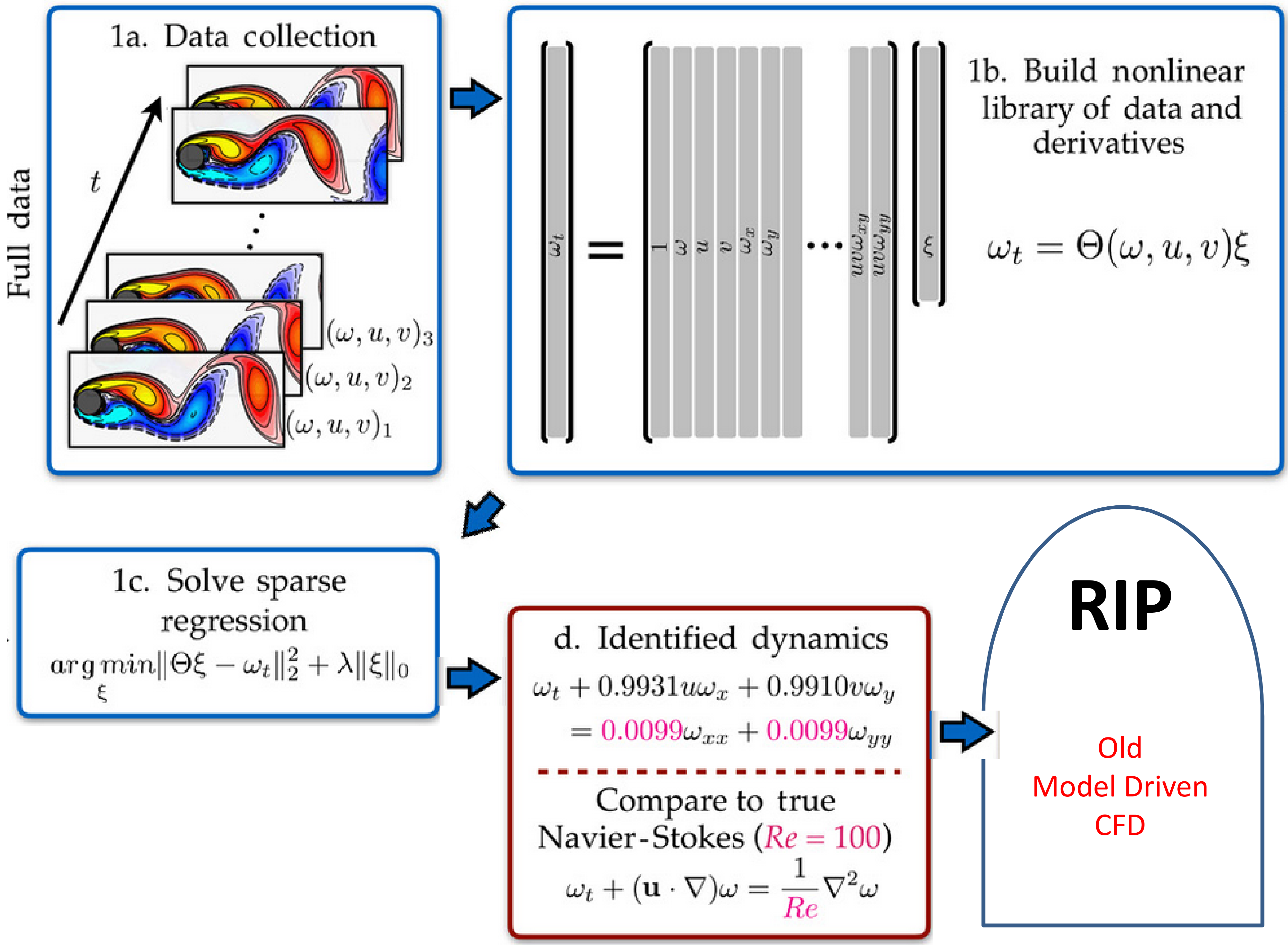
RIP

Old
Mechanical
Engineering
Department

What made it obsolete?

Electrical Vehicle

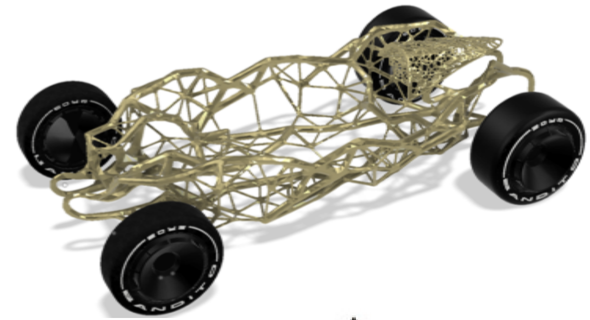
3D Printing and Laser cutting on new smart materials

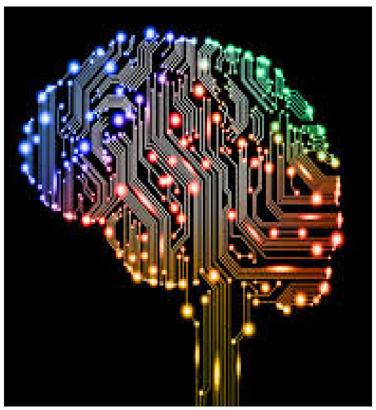

1a. Data collection
Full data
t
$(\omega, u, v)_3$
$(\omega, u, v)_2$
$(\omega, u, v)_1$
1b. Build nonlinear library of data and derivatives
ω_t
1 ω u v ω_x ω_y $\cdots$ $uv\omega_{xy}$ $uv\omega_{yy}$
ξ
$\omega_t = \Theta(\omega, u, v)\xi$
1c. Solve sparse regression
$\arg\min_{\xi} \|\Theta\xi - \omega_t\|_2^2 + \lambda\|\xi\|_0$
d. Identified dynamics
$\omega_t + 0.9931u\omega_x + 0.9910v\omega_y = 0.0099\omega_{xx} + 0.0099\omega_{yy}$
Compare to true Navier-Stokes ($Re = 100$)
$\omega_t + (\mathbf{u}\cdot\nabla)\omega = \frac{1}{Re}\nabla^2\omega$
RIP
Old
Model Driven
CFD

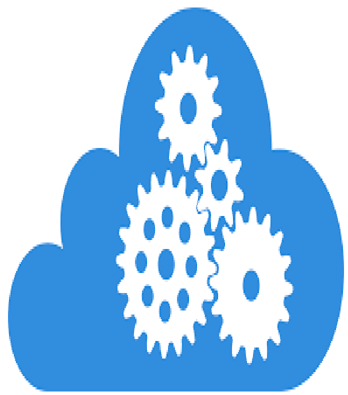
Design and manufacturing in the New Age

RIP

Old Manufacturing Engineering Department

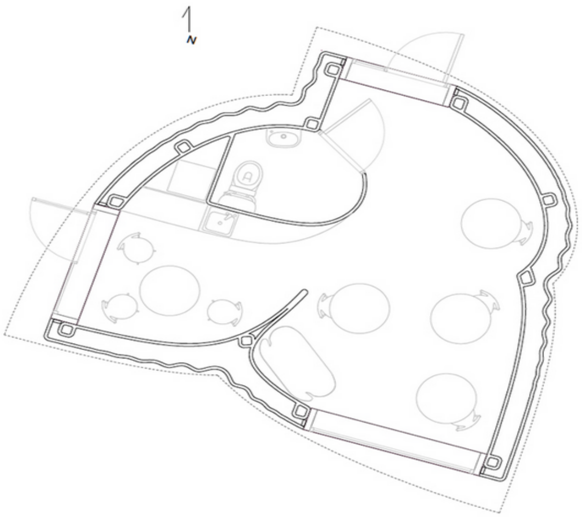






3D printing

One Human + Artificial Intelligence Algorithms + Unlimited Cloud Computing Power = 100s to 1,0000 Of Design Options



3D Printing of House with Smart materials

RIP

Old Civil Engineering Department

Learning to Plan Chemical Syntheses

Marwin Segler[♣,∇] Mike Preuss[H] Mark P. Waller[♠ℏ]
[♣]Institute of Organic Chemistry
[∇]Center for Multiscale Theory and Computation
[♠]Department of Physics
[ℏ]International Center for Quantum

AI-assisted computational chemistry: Predicting chemical properties with minimal expert knowledge

Garrett Goh (Pacific Northwest National Lab)
1:45pm-2:25pm Thursday, June 29, 2017

Prediction of Organic Reaction Outcomes Using Machine Learning

Connor W. Coley,[†] Regina Barzilay,[‡] Tommi S. Jaakkola,[‡] William H. Green,[*,†] and Klavs F. Jensen[*,†]

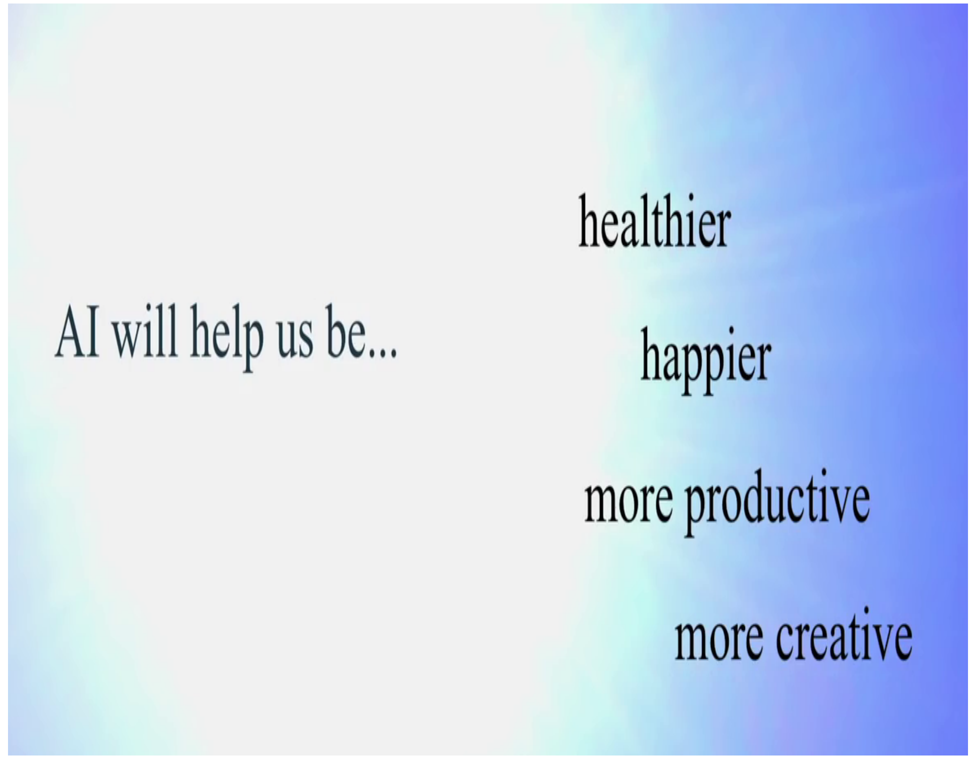
[†]Department of Chemical Engineering and [‡]Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, Massachusetts 02139, United States

RIP

Old
Chemistry
And Chemical
Engineering
Department

Quantum-chemical insights from deep tensor neural networks

Kristof T. Schütt[1], Farhad Arbabzadah[1], Stefan Chmiela[1], Klaus R. Müller[1,2] & Alexandre Tkatchenko[3,4]



Let it be true

Thank you

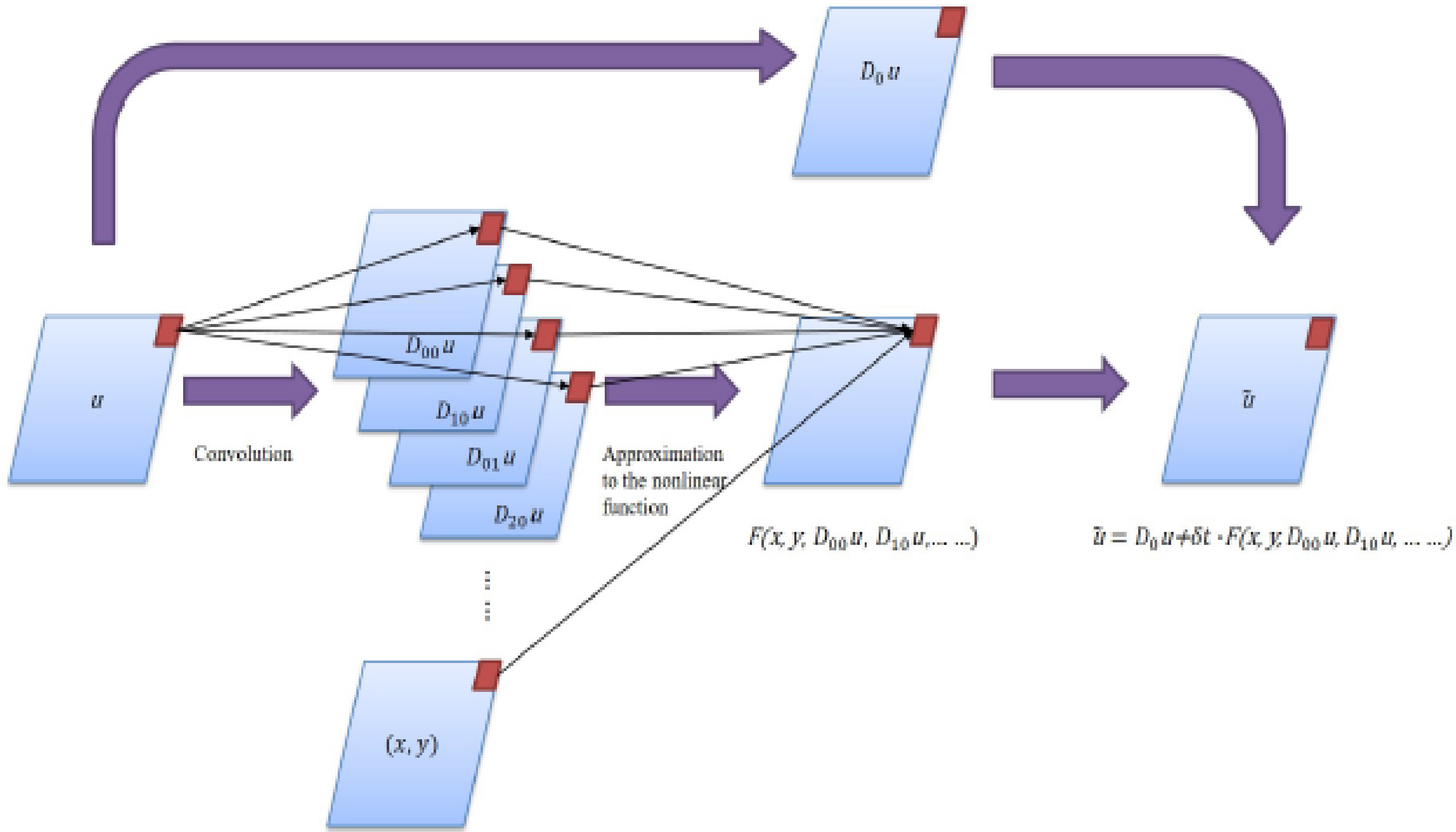


Figure 1: The schematic diagram of a δt-block.

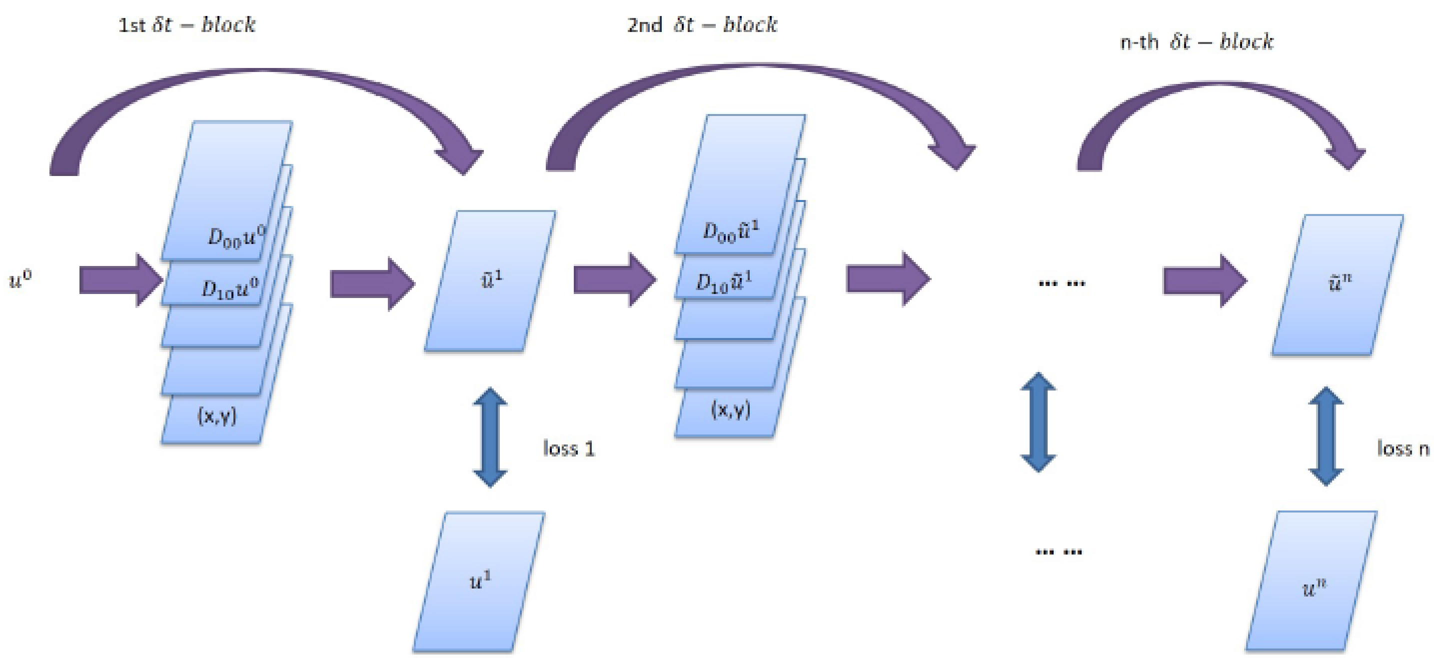
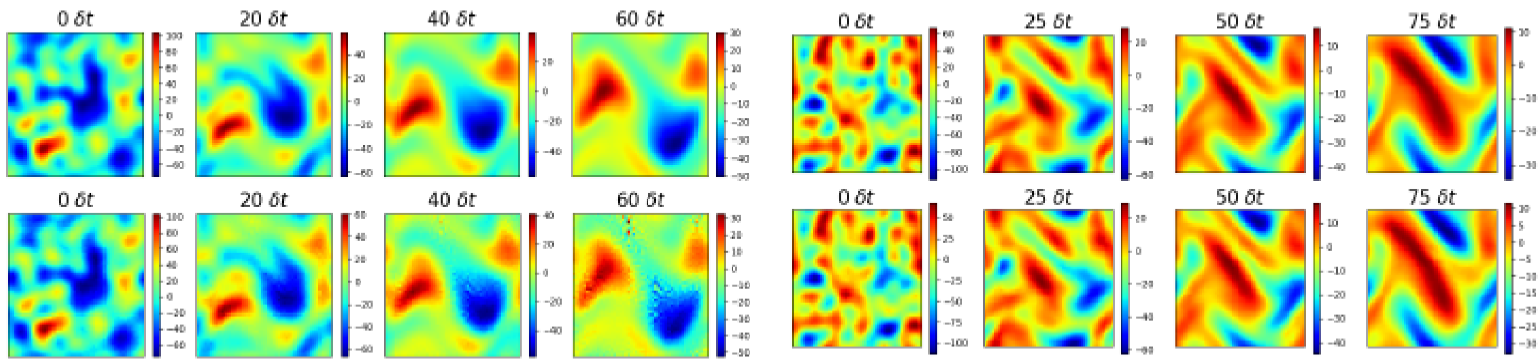


Figure 2: The schematic diagram of the PDE-Net: multiple δt-blocks.



(a) 5×5 filters.

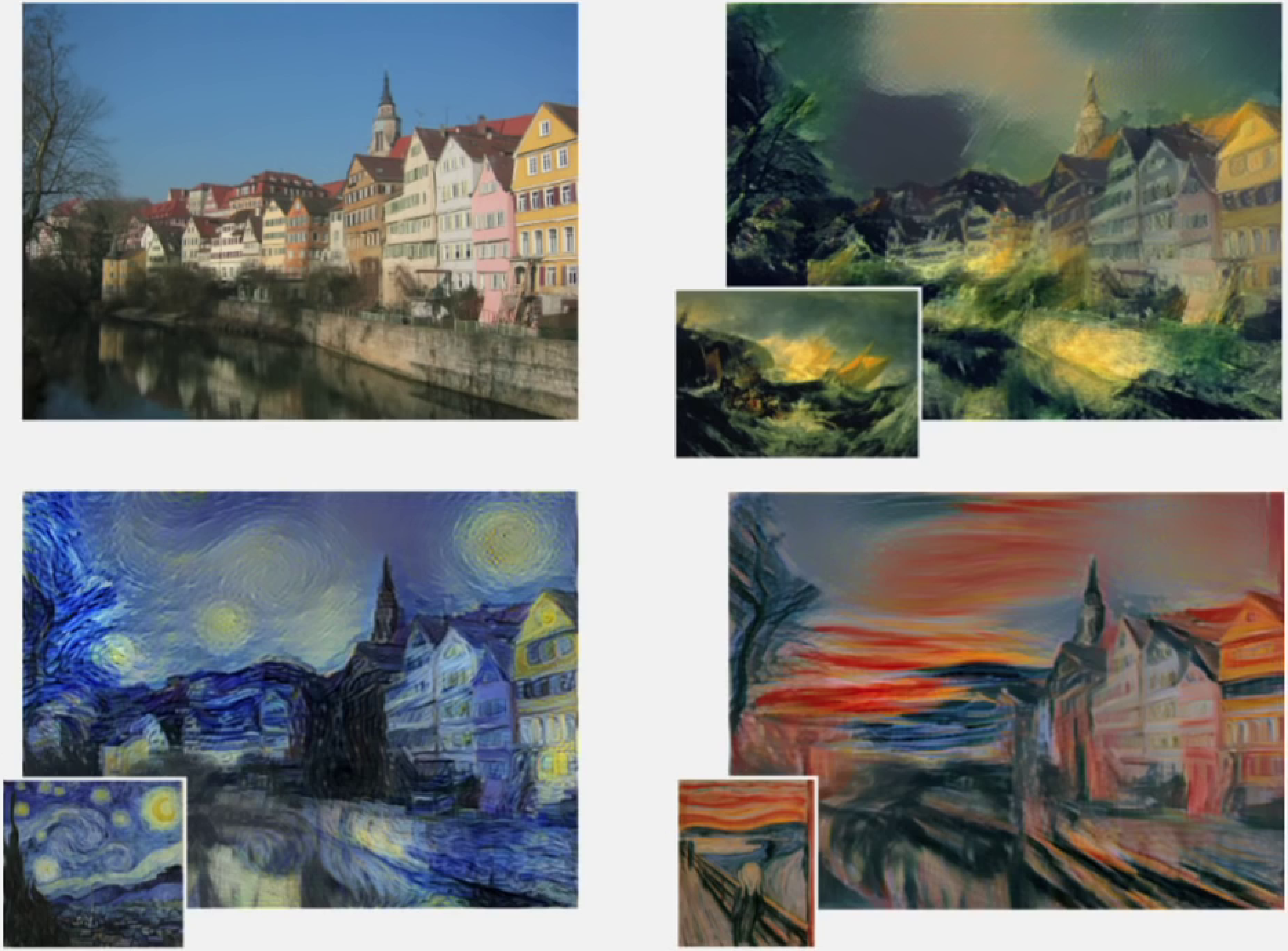
(b) 7×7 filters.

Figure 5: Images of the true dynamics and the predicted dynamics. The first row shows the images of the true dynamics. The second row shows the images of the predicted dynamics using the PDE-Net having 3 δt-blocks with 5×5 and 7×7 filters. Time step $\delta t = 0.01$.

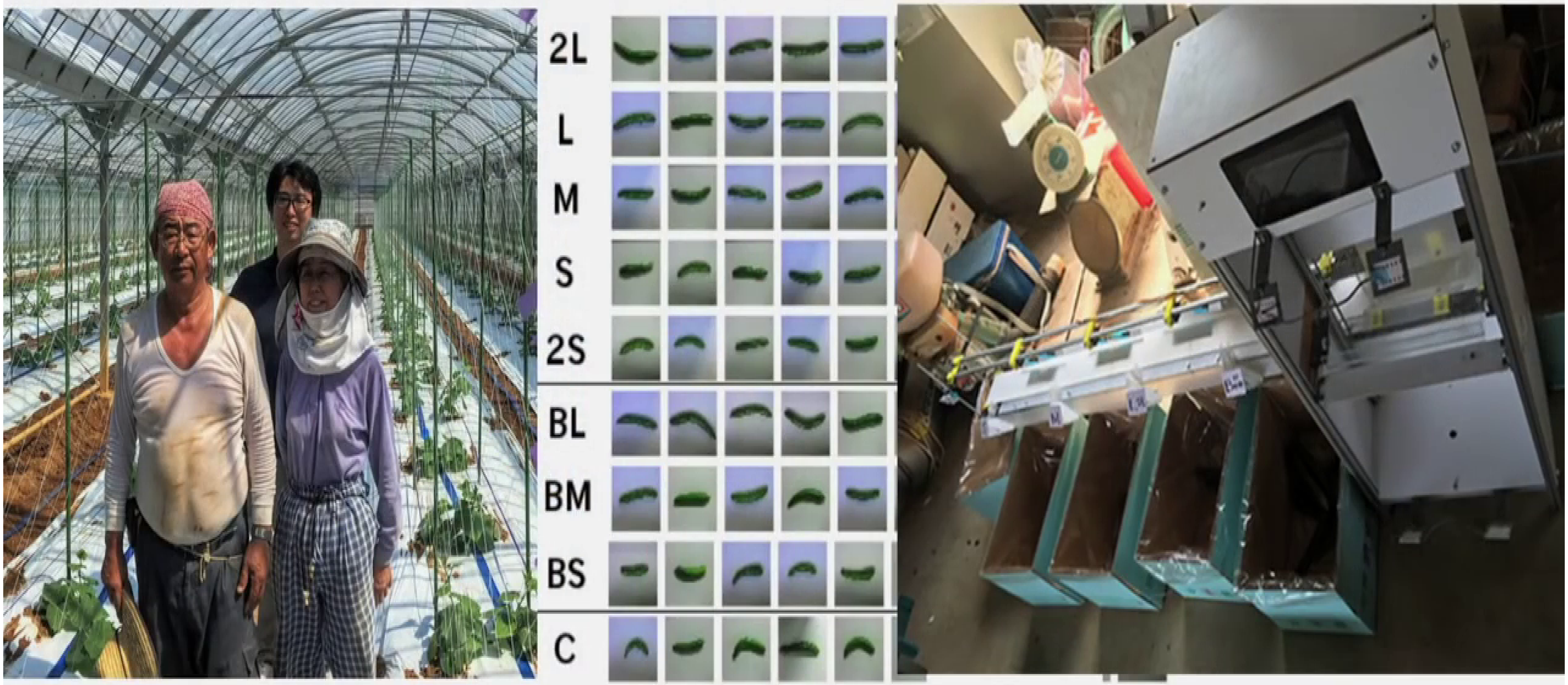
sophisticated creative tools

Take one photograph and a hand-painted picture. Then Merge

A Neural Algorithm of Artistic Style, Gatys, Ecker, & Bethge



How a Japanese cucumber farmer is using deep learning and TensorFlow



cloud.google.com/blog/big-data/2016/08/how-a-japanese-cucumber-farmer-is-using-deep-learning-and-tensorflow

AI software better at identification than experts

What Could the Right Side Be?

Limited by your imagination

$$\Theta(\mathbf{X}) = \begin{bmatrix} | & | & | & | & & | & | & | & | & \\ \mathbf{1} & \mathbf{X} & \mathbf{X}^{P_2} & \mathbf{X}^{P_3} & \cdots & \sin(\mathbf{X}) & \cos(\mathbf{X}) & \sin(2\mathbf{X}) & \cos(2\mathbf{X}) & \cdots \\ | & | & | & | & & | & | & | & | & \end{bmatrix}$$

2nd degree polynomials

$$\mathbf{X}^{P_2} = \begin{bmatrix} x_1^2(t_1) & x_1(t_1)x_2(t_1) & \cdots & x_2^2(t_1) & x_2(t_1)x_3(t_1) & \cdots & x_n^2(t_1) \\ x_1^2(t_2) & x_1(t_2)x_2(t_2) & \cdots & x_2^2(t_2) & x_2(t_2)x_3(t_2) & \cdots & x_n^2(t_2) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ x_1^2(t_m) & x_1(t_m)x_2(t_m) & \cdots & x_2^2(t_m) & x_2(t_m)x_3(t_m) & \cdots & x_n^2(t_m) \end{bmatrix}$$

Three things upfront

- Ax=b
- Model Selection
- Dimensionality Reduction

Model Selection

- 1950s KL (Kullback-Leibler) divergence
- Early 70s Akaike Information Criteria
- 78 Bayes Information Criteria (G. Schwarz)
- BIC/AIC limited number of models

Automation impact: By 2021, one in four job cuts may be from India

IT, IT-enabled services and security services, followed by banking, will be the first sectors to feel the heat, says HR solutions firm PeopleStrong

“These job cuts due to automation will not happen immediately, but the impact will become prominent by 2020. The change has started, with companies introducing bots for customer service, managing warehouses, etc.,”

Japanese firm to open world's first robot-run farm

Spread (a vegetable producing company) says it will open the fully automated farm with robots handling almost every step of the process