Three Parts

1. What are Artificial Intelligence, Machine Learning, and Deep Learning?
2. Deep Learning
3. Probabilistic Circuits and the Automated Scientist

A Short History of ~~Time~~ Artificial Intelligence, Machine Learning, and Deep Learning

Thanks to Christoph Lampert and Constantin Rothkopf for some of the slides

Kristian Kersting

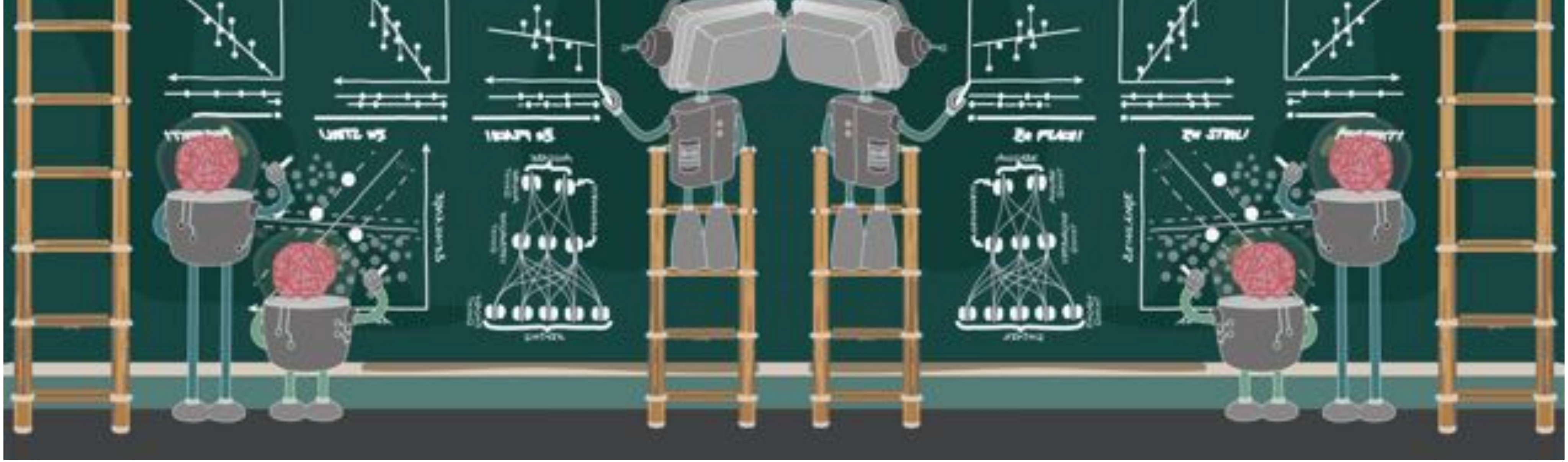


Illustration Nanina Föhr

Solving Rubik‘s Cube?

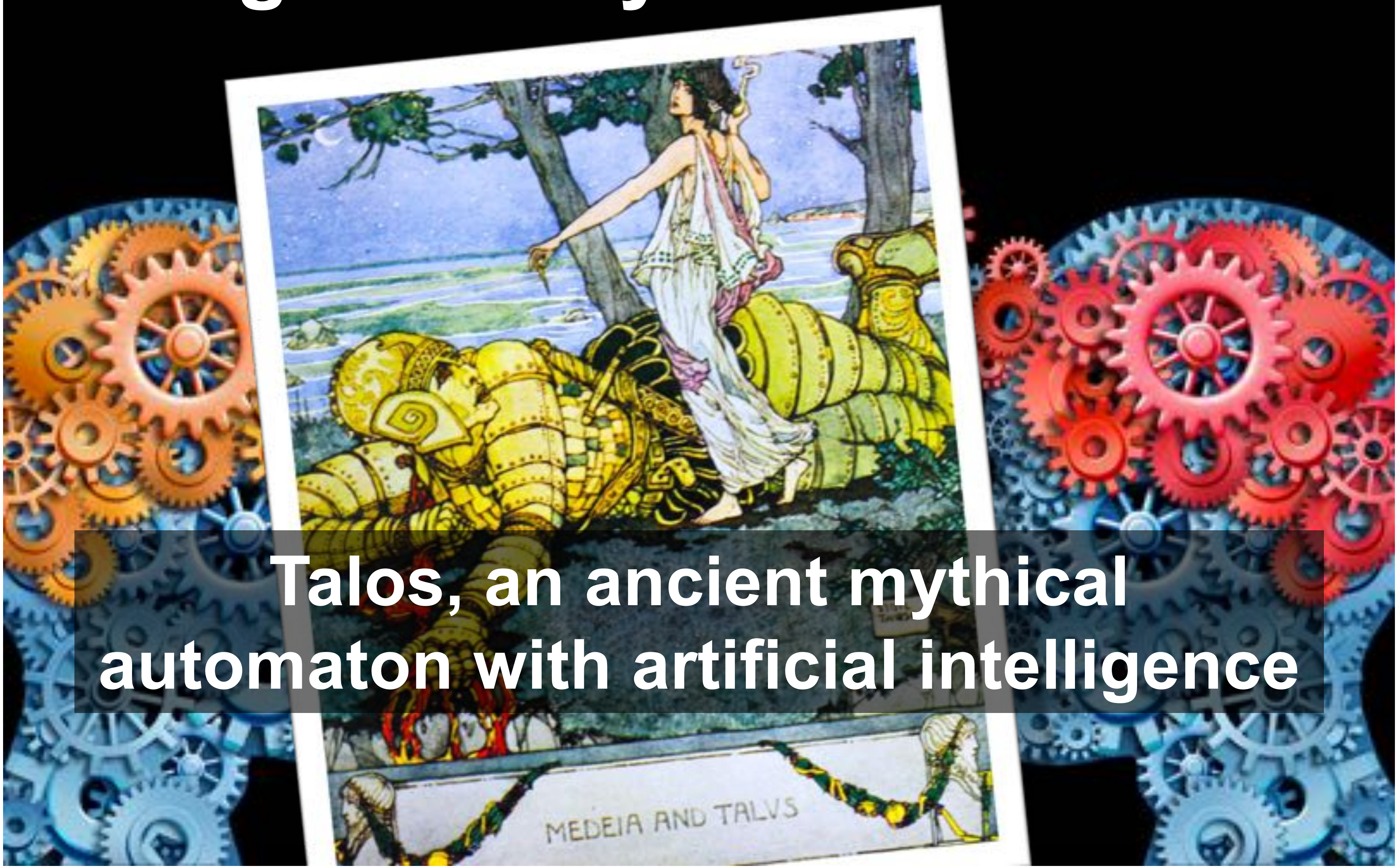
OpenAI: https://www.youtube.com/watch?v=x4O8pojMF0w

Your turn!

What do you think? Is this AI? Is this just Machine Learning? Is this at the level of humans? Is this overselling?

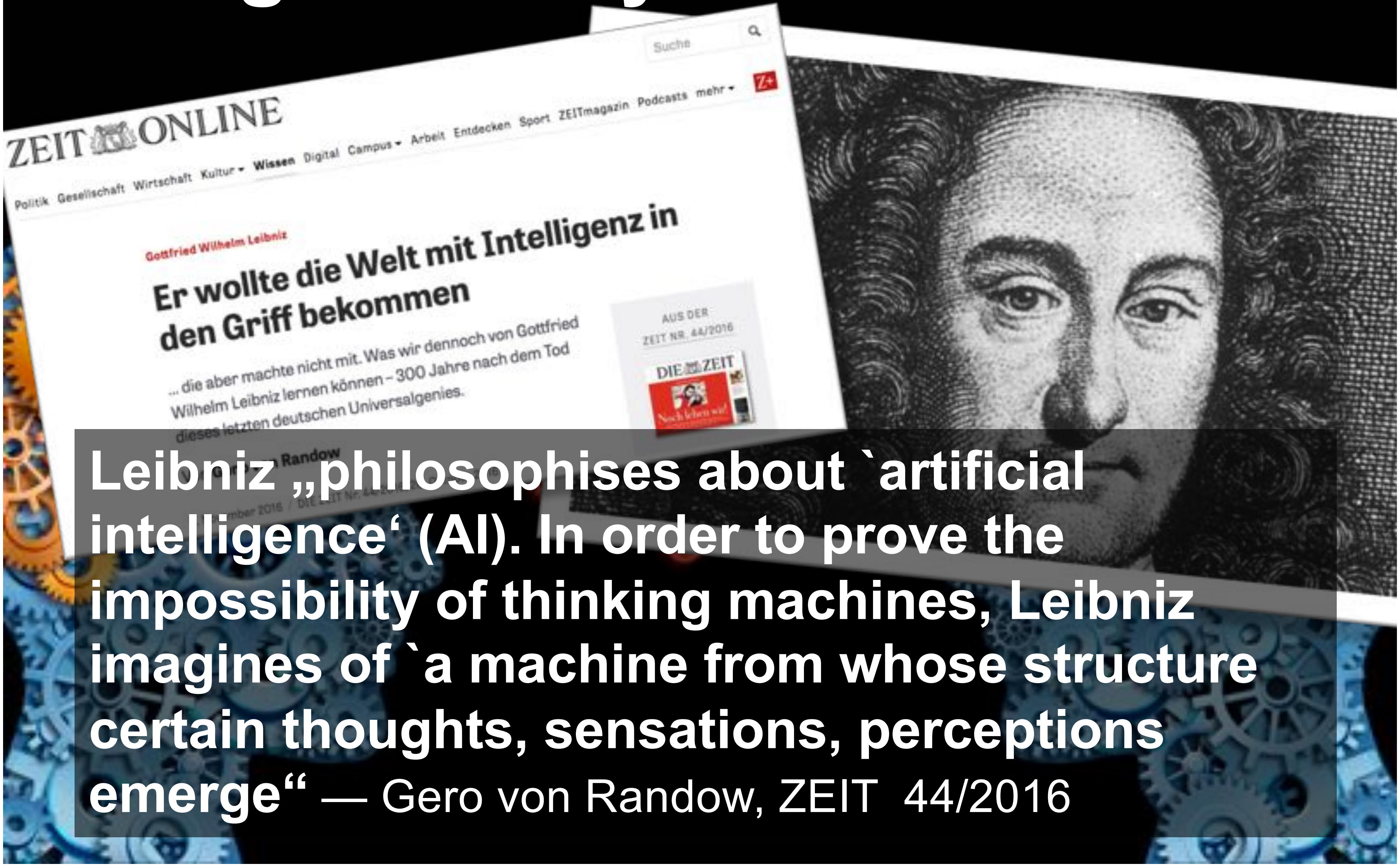
You have 5 minutes!

The dream of an artificially intelligent entity is not new



Talos, an ancient mythical automaton with artificial intelligence

The dream of an artificially intelligent entity is not new



Leibniz „philosophises about `artificial intelligence‘ (AI). In order to prove the impossibility of thinking machines, Leibniz imagines of `a machine from whose structure certain thoughts, sensations, perceptions emerge“ — Gero von Randow, ZEIT 44/2016

AI today


the INQUIRER
Artificial intelligence will create the next industrial revolution, experts claim
Artificial intelligence better than scientists at choosing successful embryos
'We won't waste time on treatments that won't work, so the patient should get
says clinic director
BBC NEWS
Technology
Stephen Hawking warns artificial intelligence could end mankind
"Humans, who are limited by slow biological evolution, couldn't compete and would be
Telegraph
HOME
Lifestyle · Cars · News
Self-driving Tesla 'sav
by steering him to hos
V/S
Elon Musk
@elonmusk
I've talked to Mark about this. His understanding of the subject is limited.
SCIENTIFIC AMERICAN DECEMBER 2016
Computers Now Recognize Patterns Better Than Humans Can
An approach to artificial intelligence that enables computers to recognize visual patterns better than humans are able to do

AI today


THE ECONOMIC IMPACT OF
ARTIFICIAL INTELLIGENCE
NORTH AMERICA
$3.7 TR
NORTHERN EUROPE
$1.8 TR
$0.7 TR
SOUTHERN EUROPE
CHINA
$7 TR
LATIN AMERICA
$0.5 TR
REST OF WORLD
$1.2 TR
$0.9 TR
DEVELOPED ASIA
Projected Global Economic Effects of AI by 2030
Source: PwC

So, AI has many faces

But, what exactly is AI?

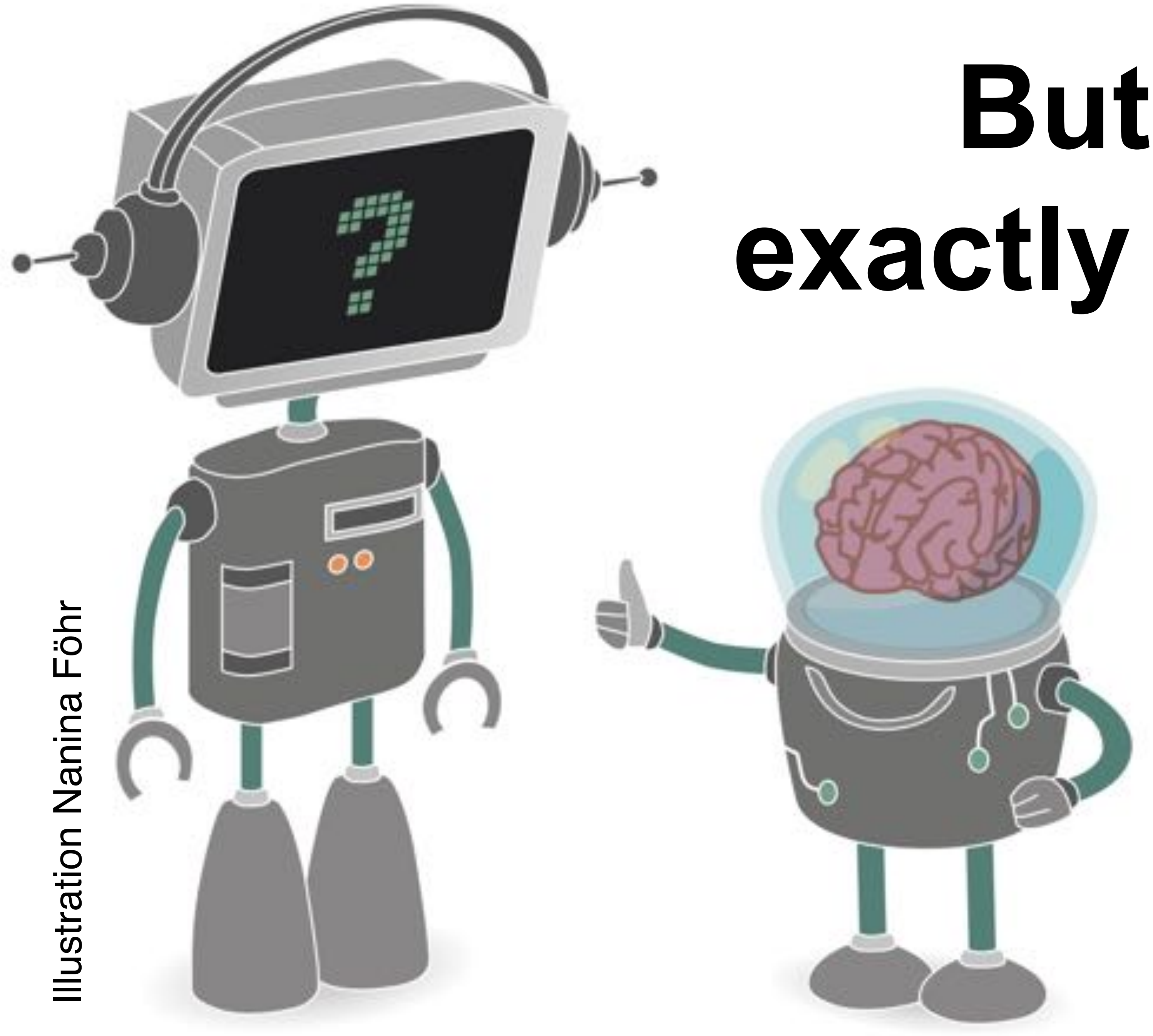


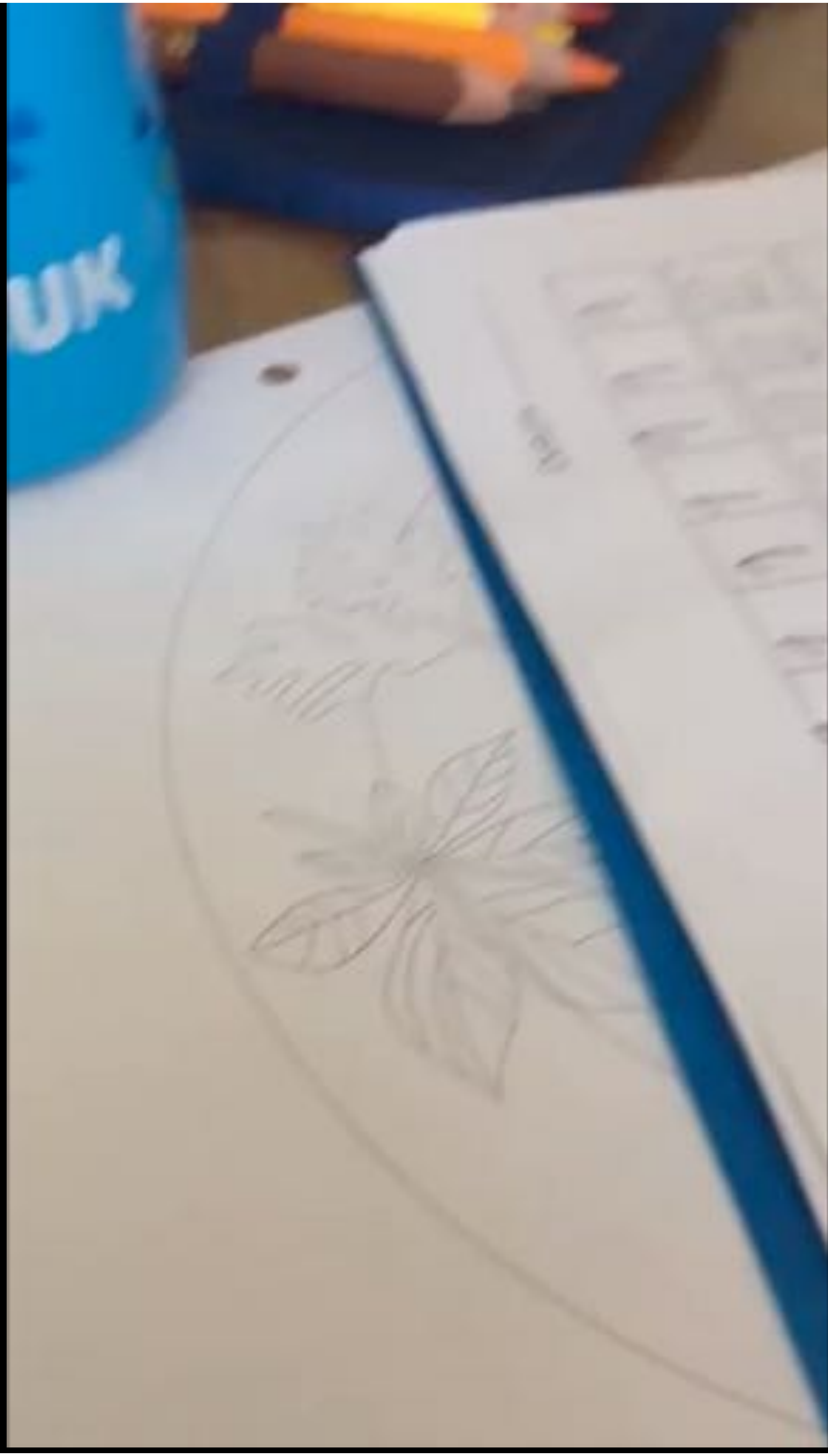
Illustration Nanina Föhr

Your turn!

What do you think is AI?

You have 5 minutes!

Humans are considered to be smart



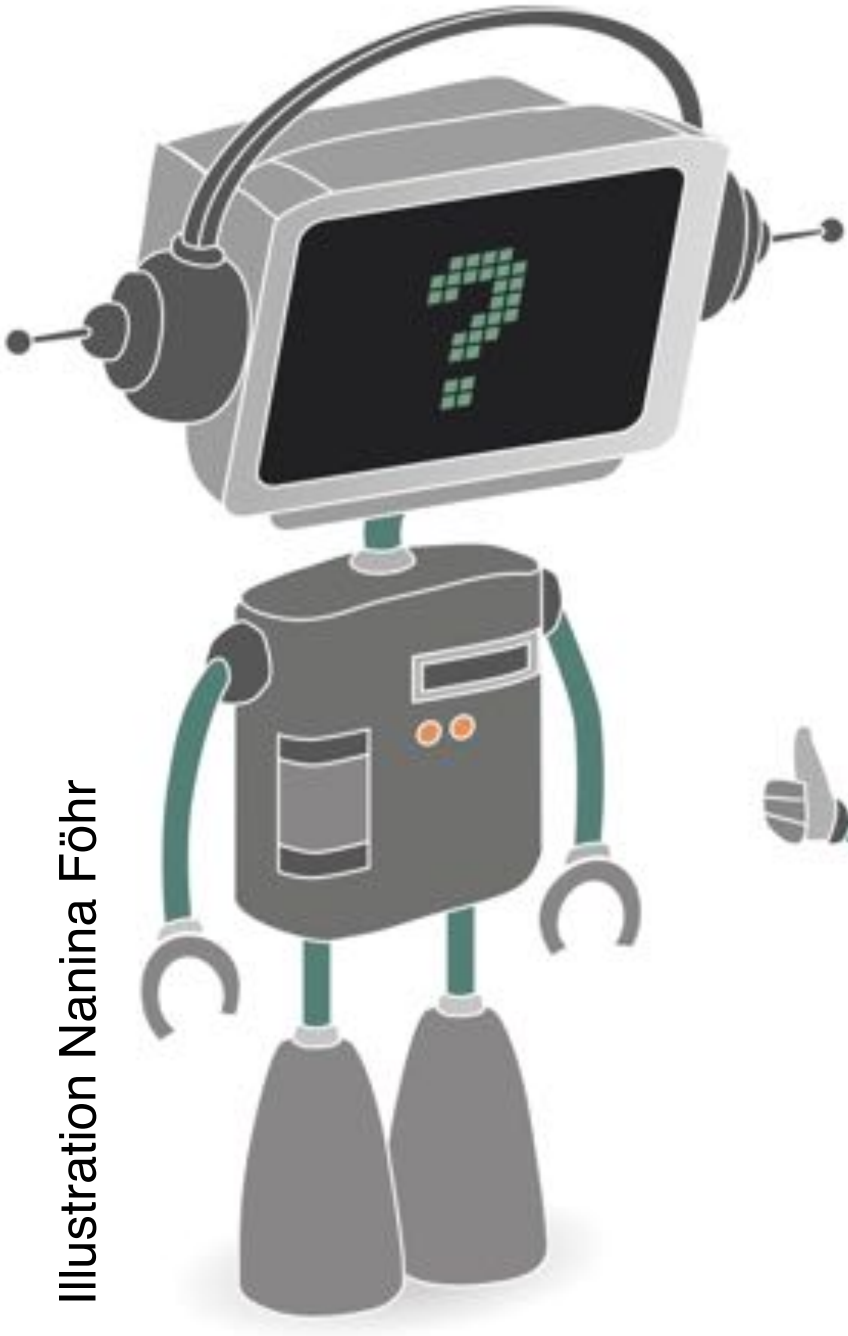
https://www.youtube.com/watch?v=XQ79UUIOeWc


Are flies smart?

What about orangutans?

Intelligence has many qualities.

It is difficult to directly capture/measure it.



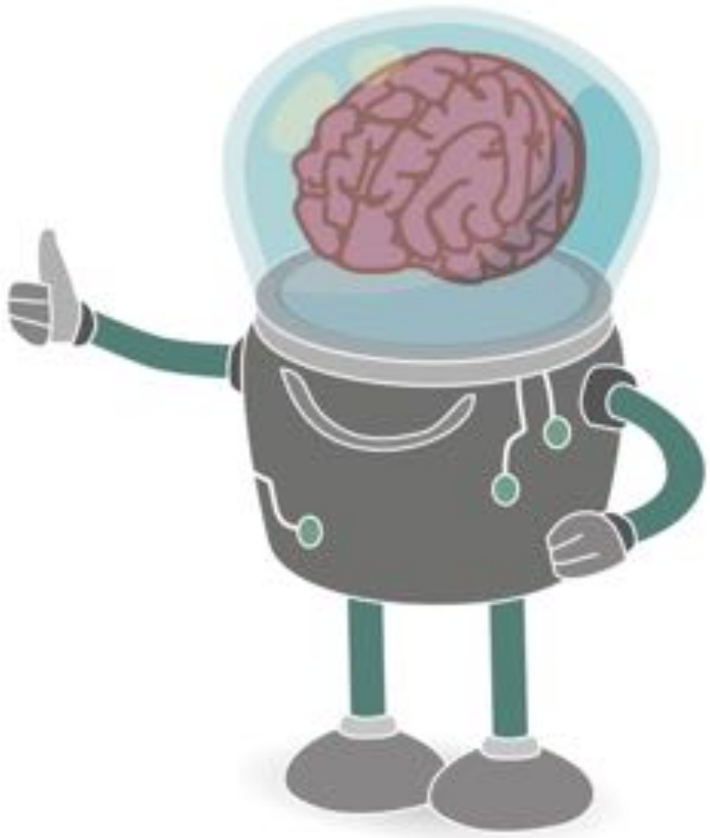


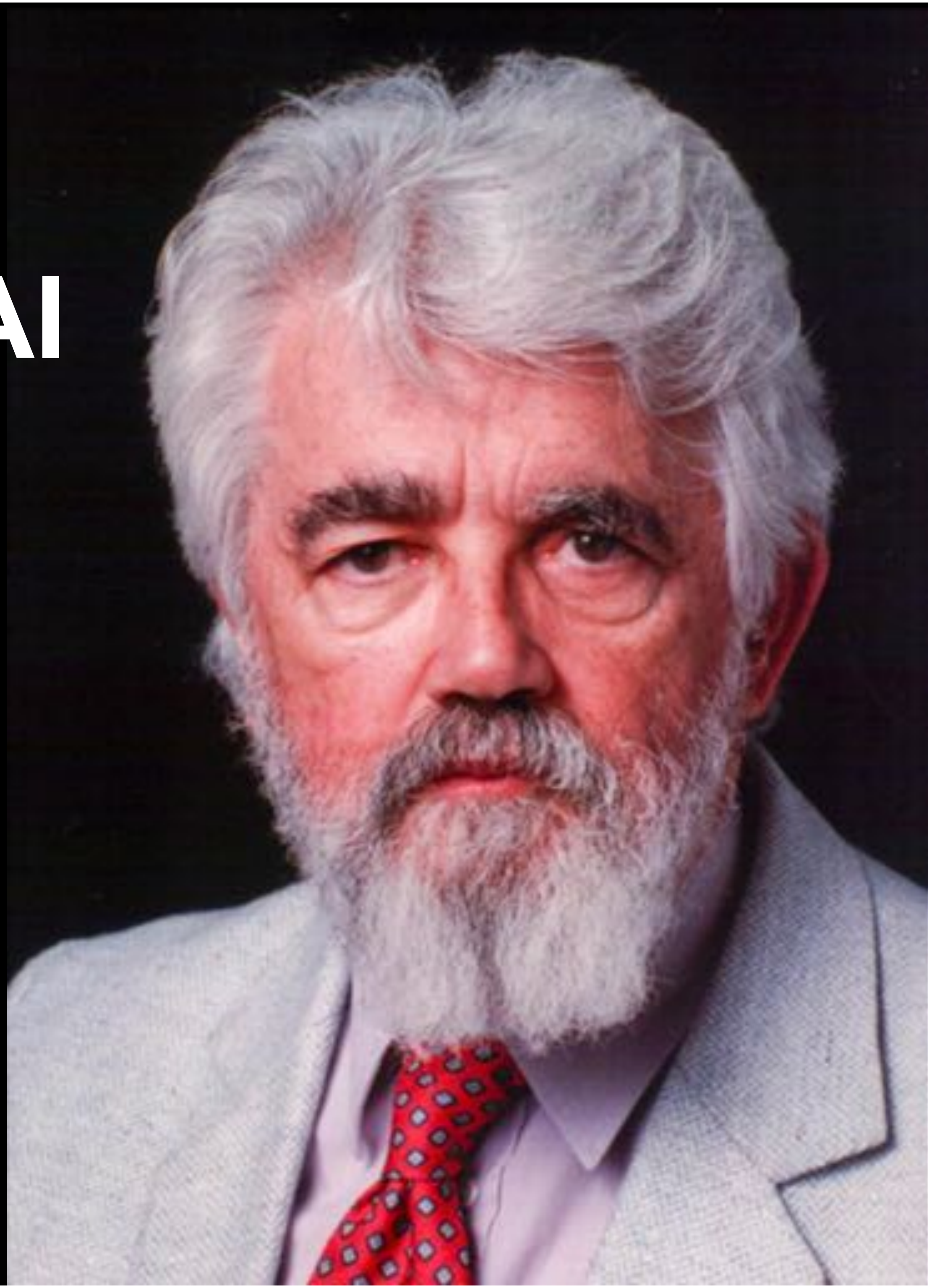
Illustration Nanina Föhr

The Definition of AI

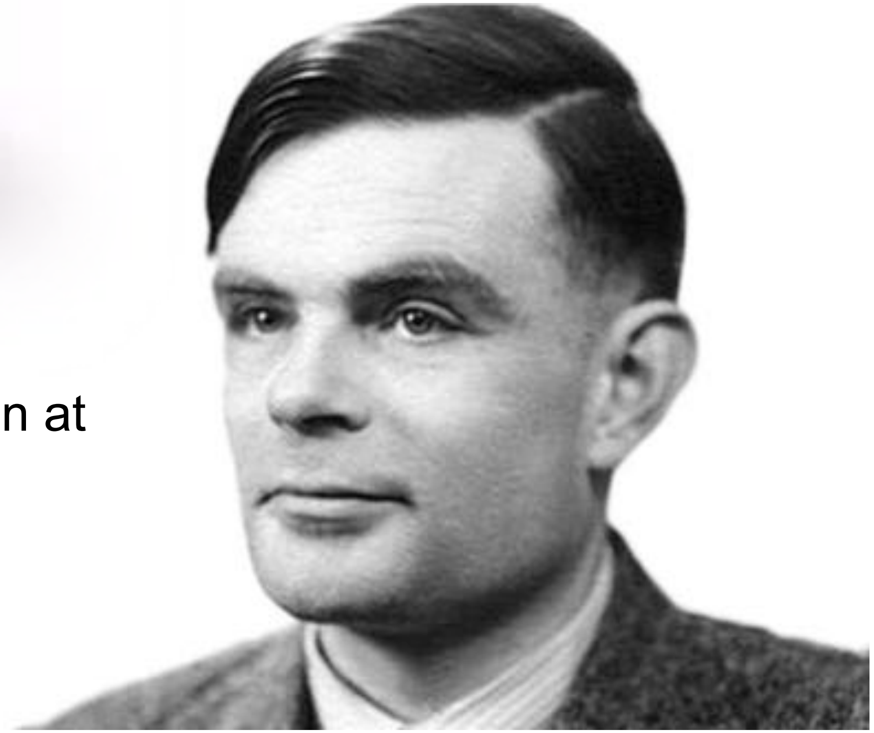
„the science and engineering of making intelligent machines, especially intelligent computer programs.

It is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable.“

- John McCarthy, Stanford (1956), coined the term AI, Turing Awardee



Turing Award =
Nobel Prize for Computing



Named after Alan Turing, a British mathematician at the University of Manchester. Turing is often credited as being the key founder of theoretical computer science and AI.

AI wants to build intelligent computer programs. How do we do this?

We use algorithms:
unambiguous specifications of how to solve a class of problems – in finite time.

Always follow the right-hand path. If you reach a dead-end, go back to the last choice point and take the next unexplored path to the right.

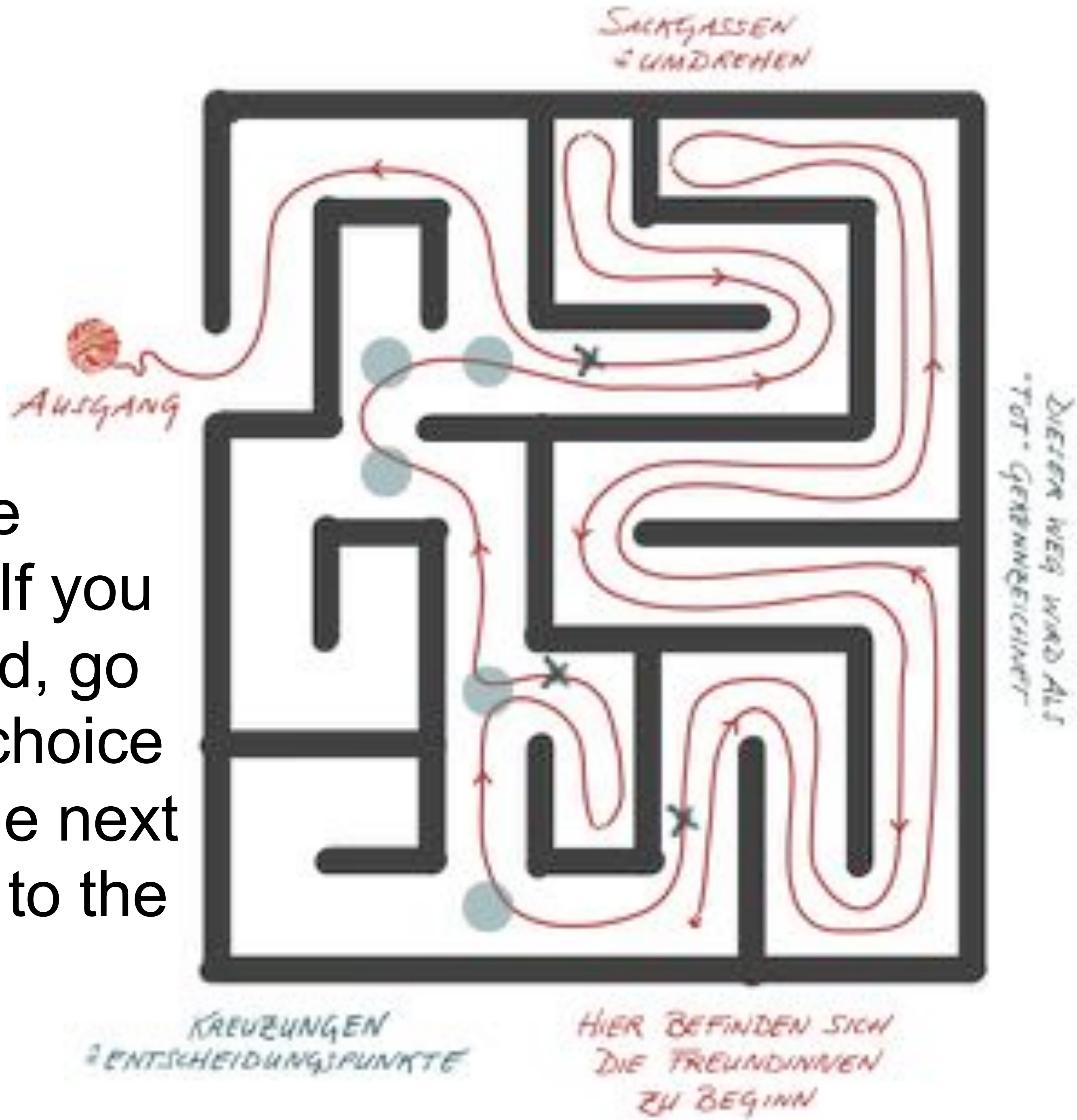


Illustration Nanina Föhr


Think of it as a recipe!

Learning

Thinking

Planning

AI = Algorithms for …

Vision

Behaviour

Reading

Machine Learning

the science "concerned with the question of how to construct computer programs that automatically improve with experience”

- Tom Mitchell (1997) CMU

Deep Learning

a form of machine learning that makes use of artificial neural networks

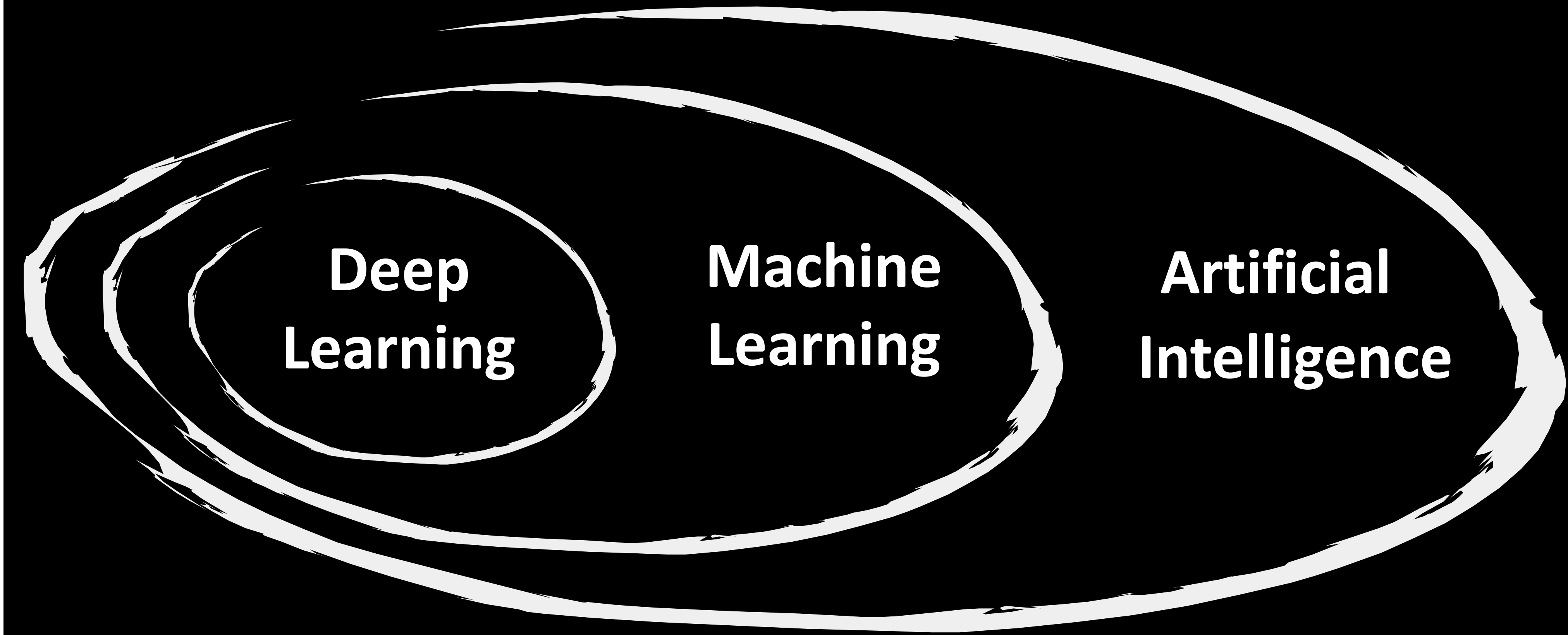
Geoffrey Hinton
Google
Univ. Toronto (CAN)

Yann LeCun
Facebook (USA)

Yoshua Bengio
Univ. Montreal (CAN)

Turing Awardees 2019

Overall Picture



Your turn?

Which examples for AI do you know? Where do you think ML is used? Do you know an example for ML that is not DL?

You have 5 minutes!

A closer look at
the history of AI

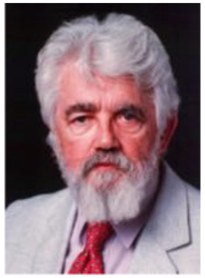
ONCE UPON A TIME

1956 Birth of AI


A Proposal for the

DARTMOUTH SUMMER RESEARCH PROJECT ON ARTIFICIAL INTELLIGENCE

We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer.



John McCarthy
Turing Award 1971

Marvin Minsky
Turing Award 1969

Allen Newell
Turing Award 1975

Herbert A. Simon
Turing Award 1975
Nobel Prize 1978

… and of Cognitive Science

Artificial Neural Networks

Learning representations by back-propagating errors

David E. Rumelhart*, Geoffrey E. Hinton† & Ronald J. Williams*

* Institute for Cognitive Science, C-015, University of California, San Diego, La Jolla, California 92093, USA
† Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Philadelphia 15213, USA

COGNITIVE SCIENCE **14**, 179–211 (1990)

Finding Structure in Time

JEFFREY L. ELMAN
University of California, San Diego

COGNITIVE SCIENCE **9**, 147–169 (1985)

A Learning Algorithm for Boltzmann Machines*

DAVID H. ACKLEY
GEOFFREY E. HINTON
Computer Science Department
Carnegie-Mellon University
TERRENCE J. SEJNOWSKI
Biophysics Department
The Johns Hopkins University

Biological Cybernetics
© by Springer-Verlag 1980
Biol. Cybernetics 36, 193–202 (1980)

Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position

Kunihiko Fukushima
NHK Broadcasting Science Research Laboratories, Kinuta, Setagaya, Tokyo, Japan

Psychological Review
1981, Vol. 88, No. 2, 135–170

Toward a Modern Theory of Adaptive Networks: Expectation and Prediction

Richard S. Sutton and Andrew G. Barto
Computer and Information Science Department
University of Massachusetts—Amherst

Psychological Review
Vol. 65, No. 6, 1958

THE PERCEPTRON: A PROBABILISTIC MODEL FOR INFORMATION STORAGE AND ORGANIZATION IN THE BRAIN [1]

F. ROSENBLATT
Cornell Aeronautical Laboratory

Artificial Neural Networks

COGNITIVE SCIENCE **4**, 179–211 (1990)

Finding Structure in Time

JEFFREY L. ELMAN
University of California, San Diego

Learning representations by back-propagating errors

David E. Rumelhart*, Geoffrey E. Hinton† & Ronald J. Williams*

* Institute for Cognitive Science C-015, University of California, San Diego, La Jolla, California 92093, USA
† Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Philadelphia 15213, USA

COGNITIVE SCIENCE, 147–169 (1985)

A Learning Algorithm for Boltzmann Machines*

DAVID H. ACKLEY
GEOFFREY E. HINTON
Computer Science Department
Carnegie-Mellon University
TERRENCE J. SEJNOWSKI
Biophysics Department
The Johns Hopkins University

Biological Cybernetics
Biol. Cybernetics 36, 193–202 (1980)

Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position

Kunihiko Fukushima
NHK Broadcasting Science Research Laboratories, Kinuta, Setagaya, Tokyo, Japan

Psychological Review
1981, Vol. 88, No. 2, 135–170

Copyright 1981 by the American Psychological Association, Inc.
0033-295X/81/8802-0135$00.75

Toward a Modern Theory of Adaptive Networks: Expectation and Prediction

Richard S. Sutton and Andrew G. Barto
Computer and Information Science Department
University of Massachusetts—Amherst

Psychological Review
Vol. 65, No. 6, 1958

THE PERCEPTRON: A PROBABILISTIC MODEL FOR INFORMATION STORAGE AND ORGANIZATION IN THE BRAIN [1]

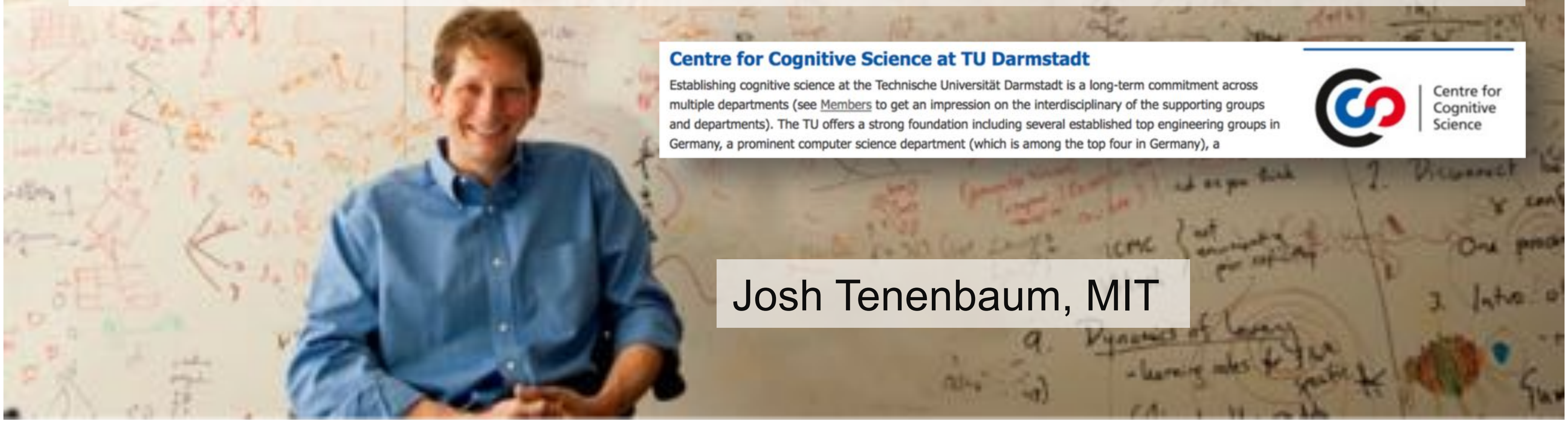
F. ROSENBLATT
Cornell Aeronautical Laboratory

slide after C. Rothkopf (TUD), after J.Tenenbaum (MIT)

Algorithms of intelligent behaviour teach us a lot about ourselves

The twin science: cognitive science

"How do we humans get so much from so little?" and by that I mean how do we acquire our understanding of the world given what is clearly by today's engineering standards so little data, so little time, and so little energy.



Centre for Cognitive Science at TU Darmstadt

Establishing cognitive science at the Technische Universität Darmstadt is a long-term commitment across multiple departments (see Members to get an impression on the interdisciplinary of the supporting groups and departments). The TU offers a strong foundation including several established top engineering groups in Germany, a prominent computer science department (which is among the top four in Germany), a

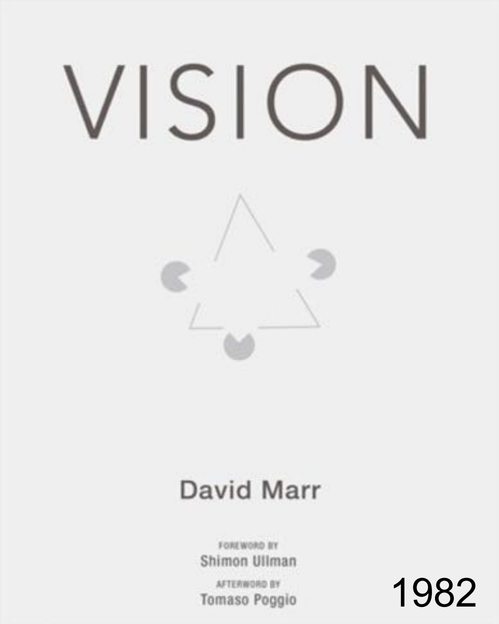
Centre for Cognitive Science

Josh Tenenbaum, MIT

Lake, Salakhutdinov, Tenenbaum, Science 350 (6266), 1332-1338, 2015

Tenenbaum, Kemp, Griffiths, Goodman, Science 331 (6022), 1279-1285, 2011

Three levels of description



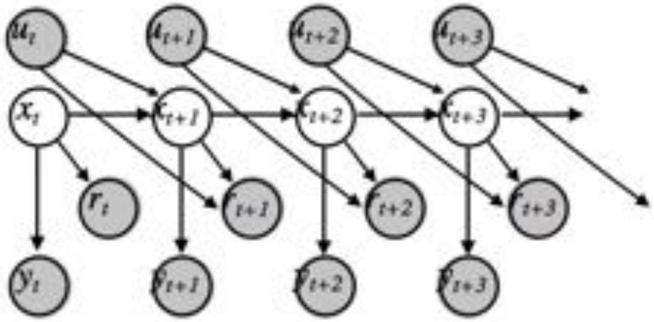
1982

Computational
Why do things work the way they work? What is the goal of the computation? What are the unifying principles?

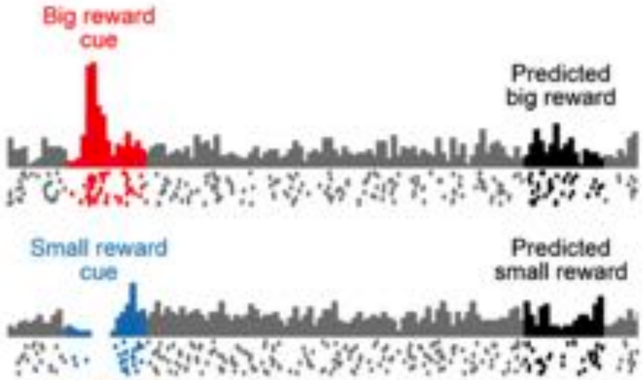
maximize:

$$R_t = r_{t+1} + r_{t+2} + \cdots + r_T$$

Algorithmic
What represetation can implement such computations? How does the choice of the representation determine the algorithm



Implementational
How can such a system be built in hardware? How can neurons carry out the computations?



slide after C. Rothkopf (TUD)

Artificial Neural Networks

Inspiration from the brain:

- **many small interconnected units (neurons)**
- **learning happens by changing the strength of connections (synapses)**
- **behavior of the whole is more than the sum of the parts**

Frank Rosenblatt (1928-1971)

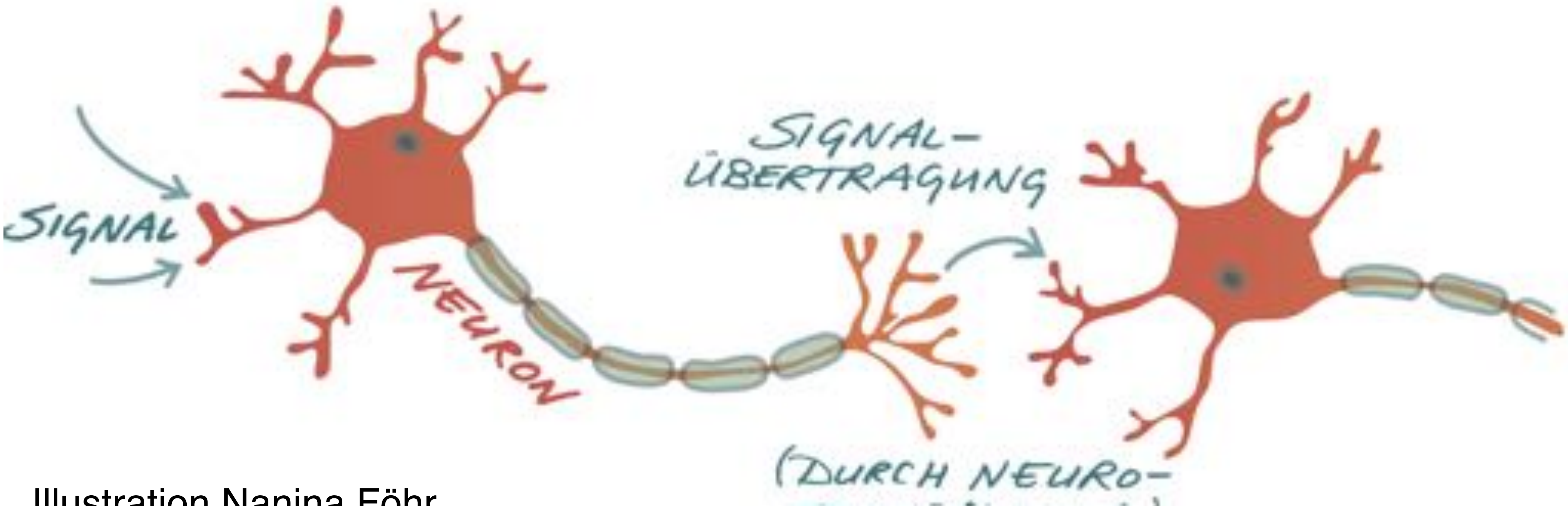


Illustration Nanina Föhr

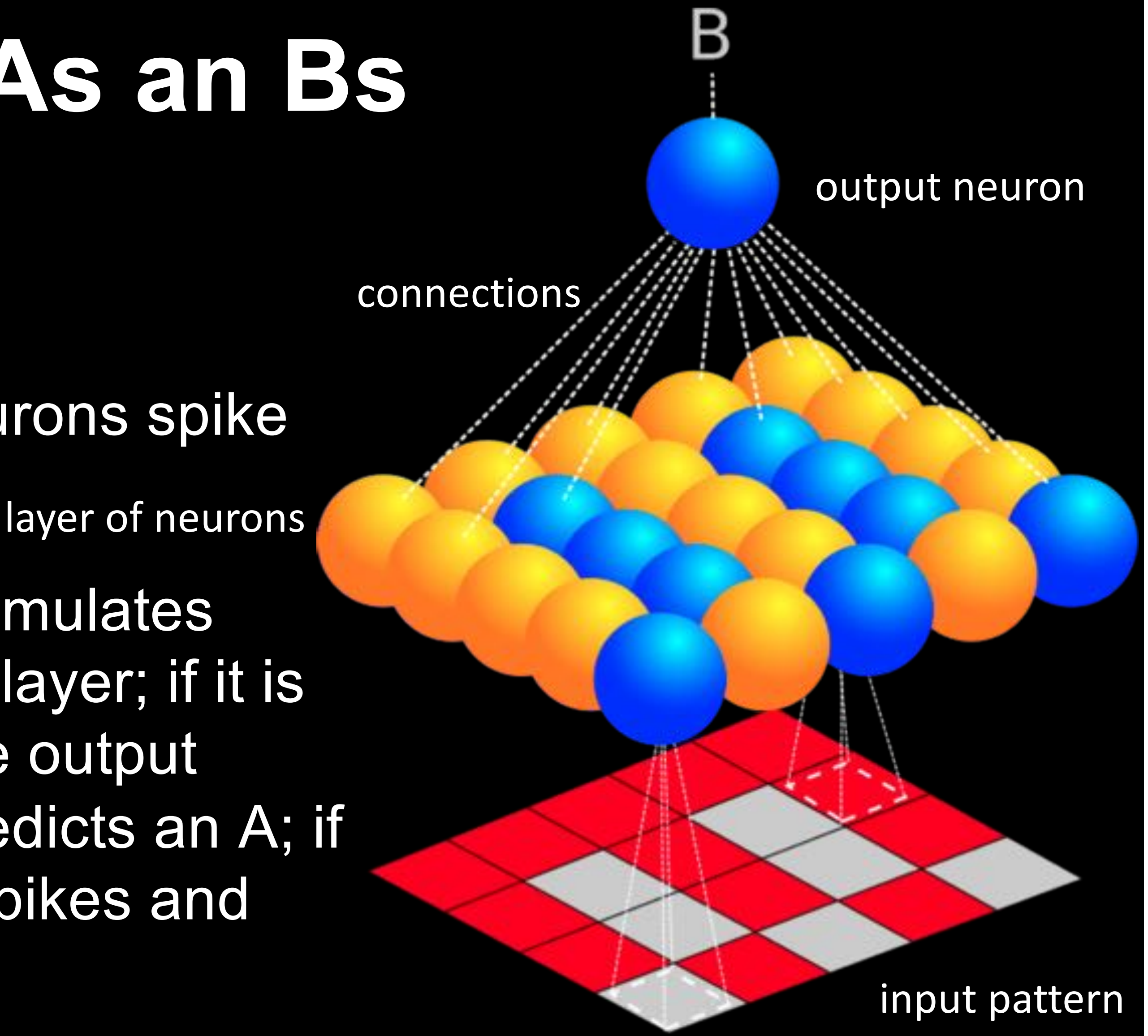
The Perceptron to distinguish As an Bs

1) present pattern

2) some first layer neurons spike

3) output neuron accumulates signals from previous layer; if it is above a threshold, the output neuron spikes and predicts an A; if not, then it does not spikes and predicts a b

4) prediction is “B”



The Perceptron Learning Algorithm

1) present pattern

2) wait for output to be produced

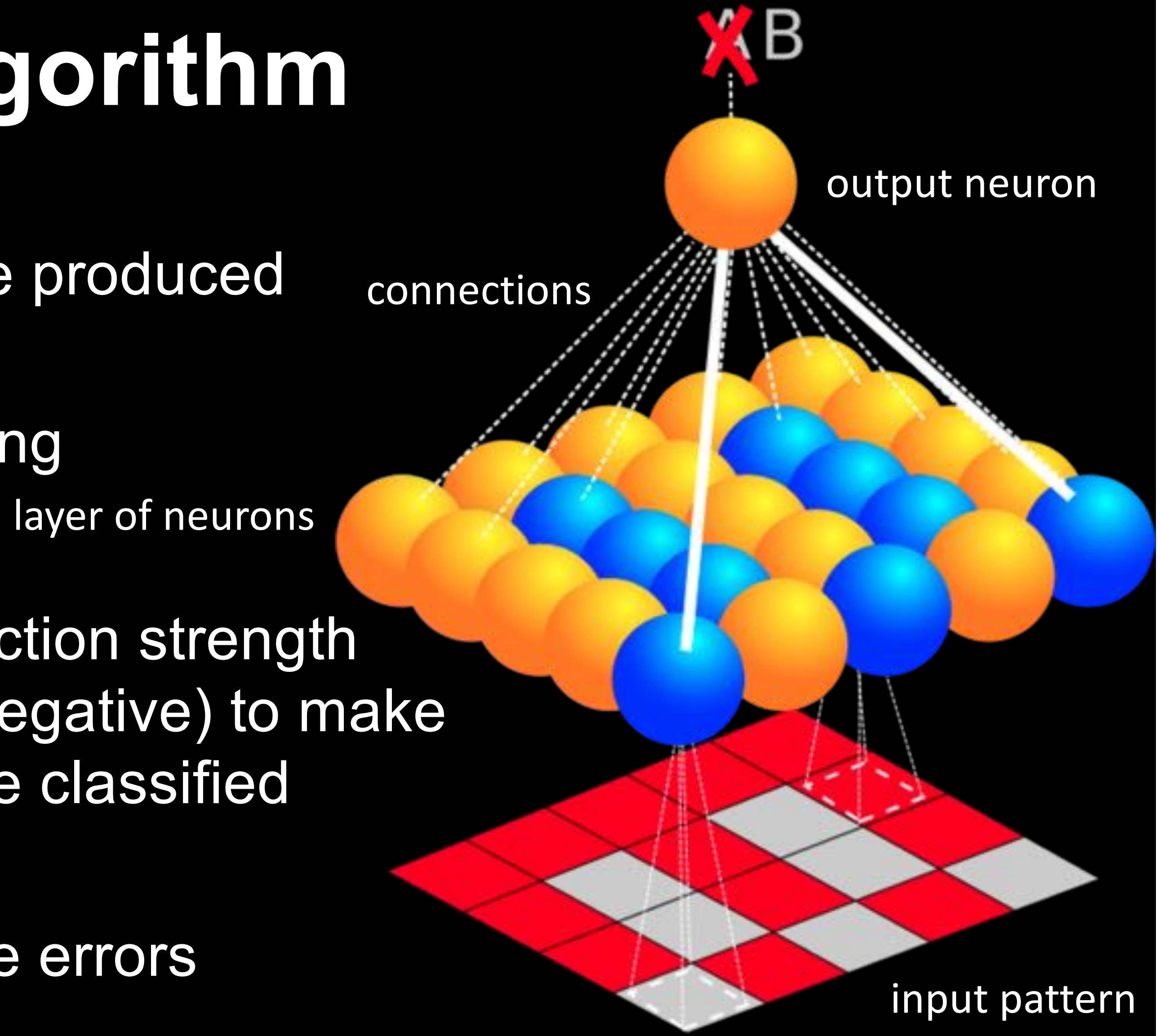
3) if output correct

- change nothing

4) if output incorrect:

- adjust connection strength (positive or negative) to make the pattern be classified correctly

5) repeat until no more errors



Artificial Neural Networks
= Stacking of many artificial neurons

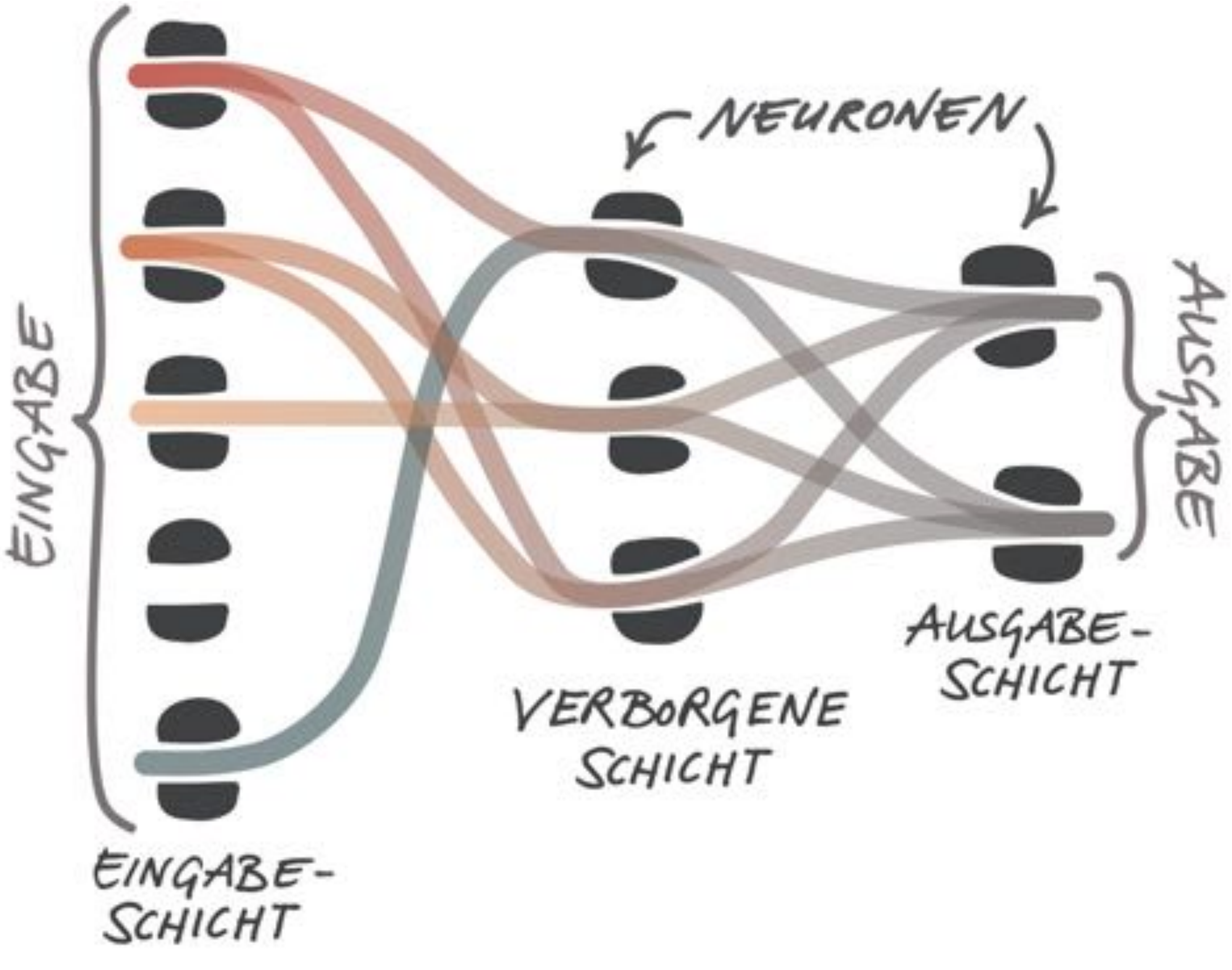
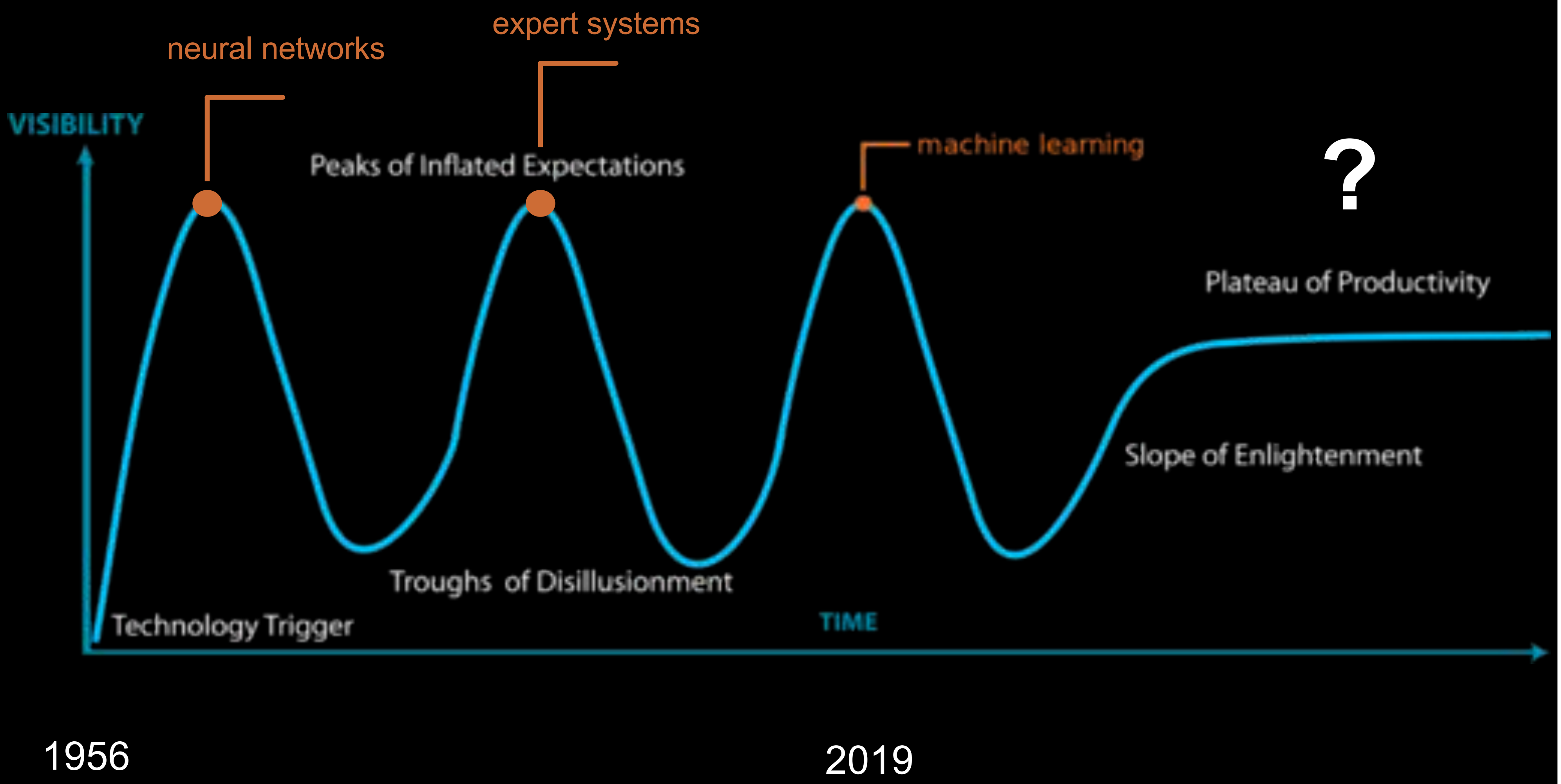
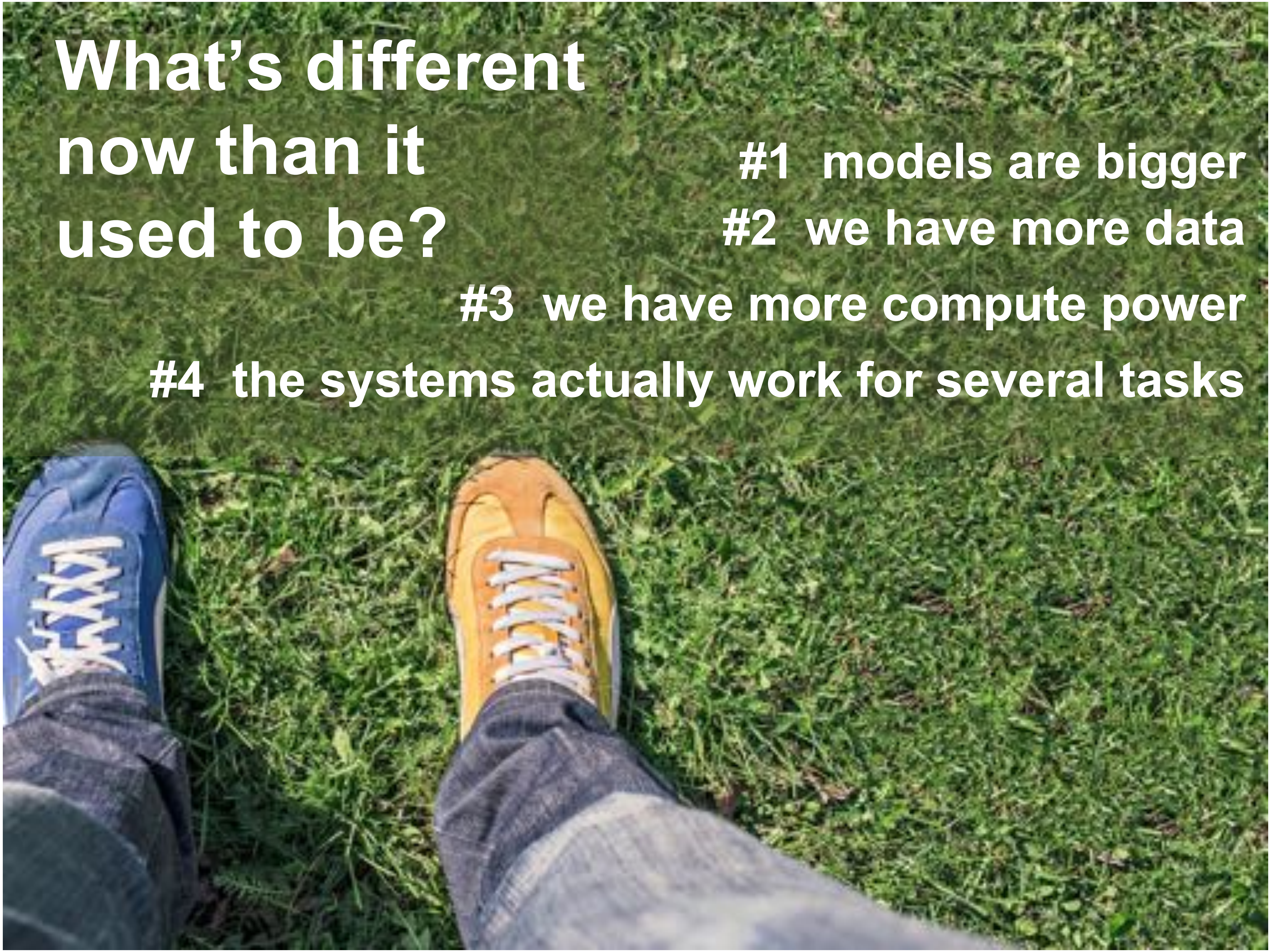


Illustration Nanina Föhr

The history of AI in a nutshell

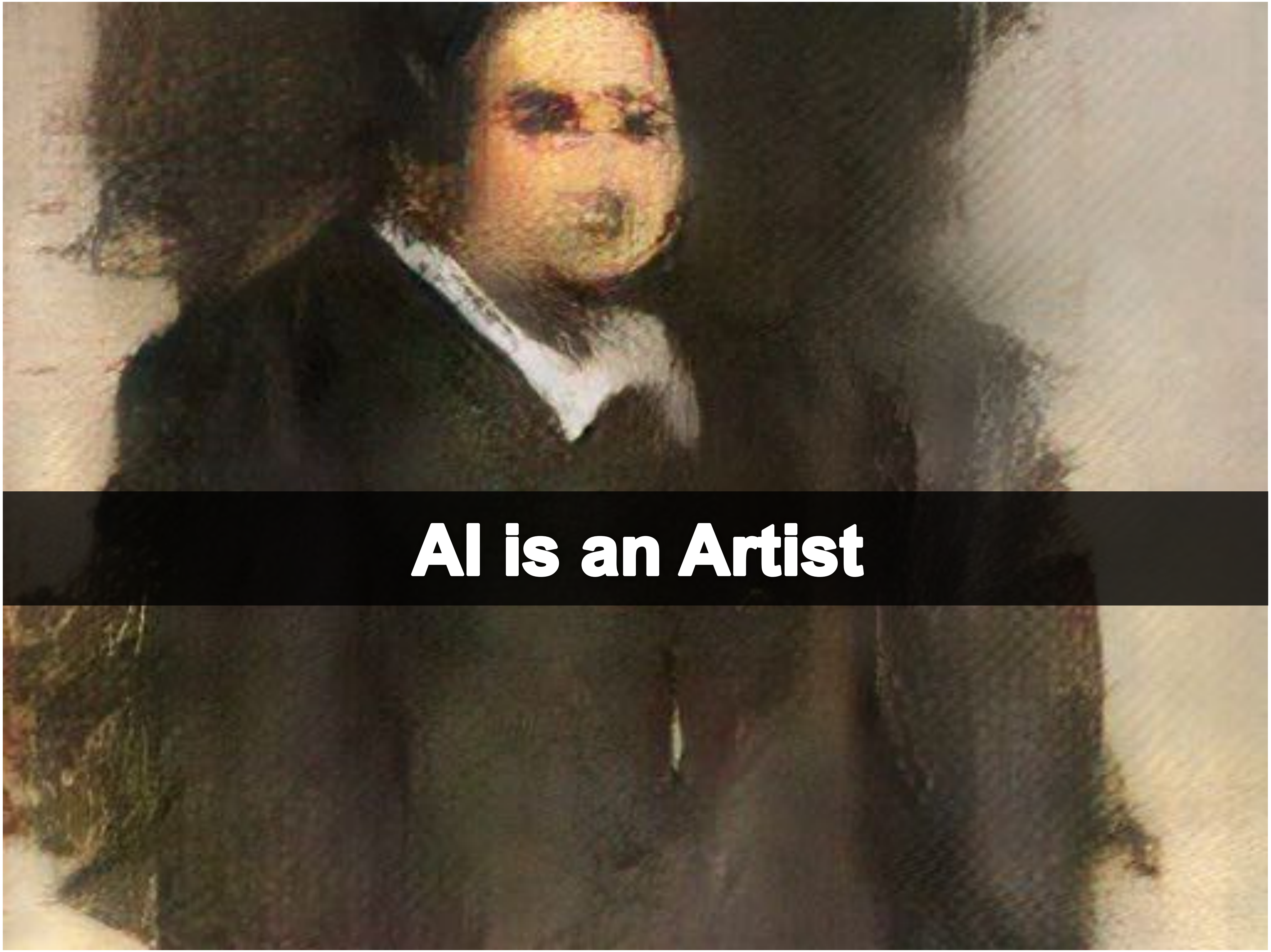



What's different now than it used to be?
#1 models are bigger
#2 we have more data
#3 we have more compute power
#4 the systems actually work for several tasks

AI drives cars


AI does the laundry

AI knows a lot


AI is an Artist

TECHNISCHE UNIVERSITÄT DARMSTADT

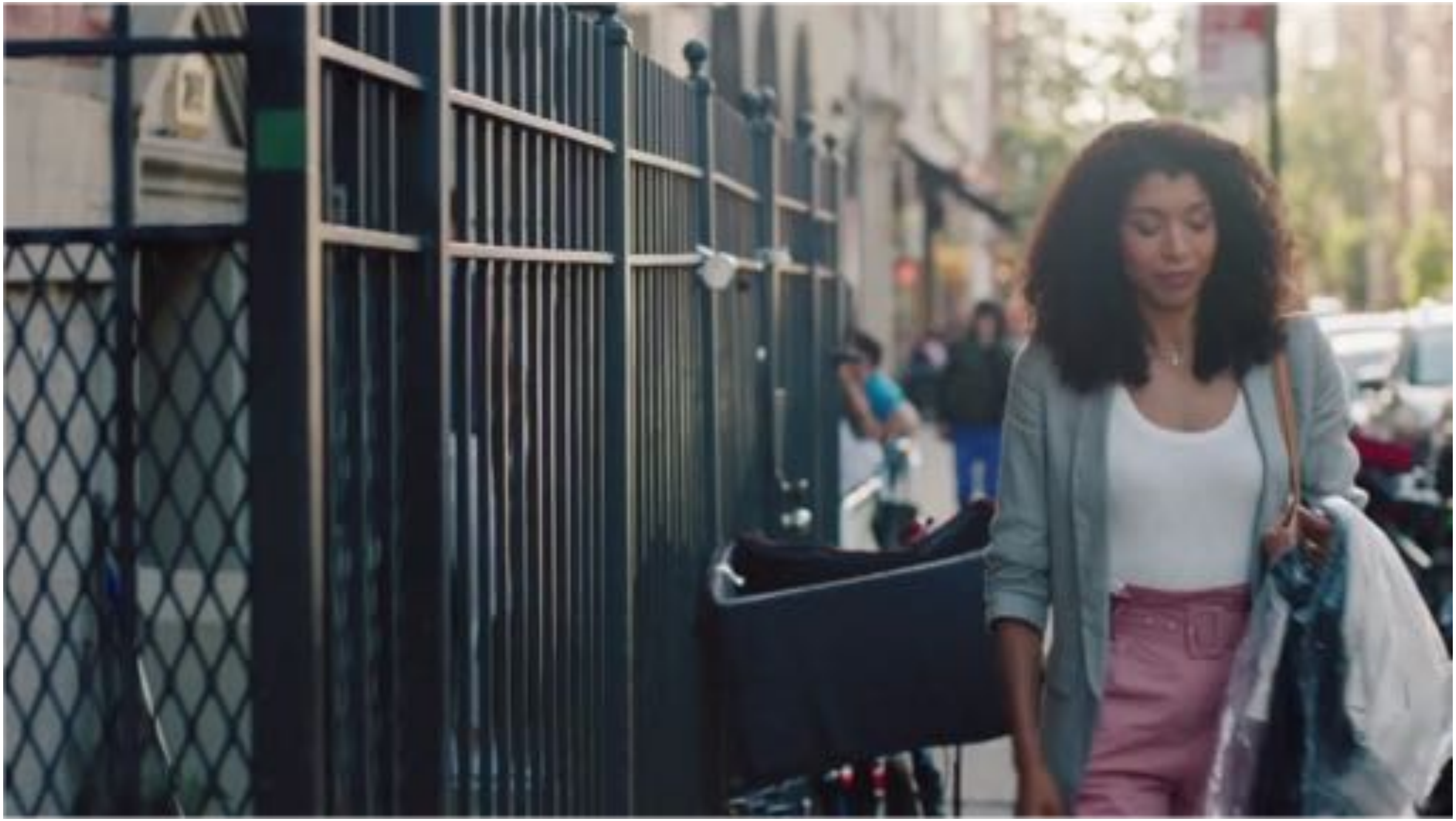
AI plays chess and GO

Schachmatt durch „CrazyAra" 19.02.2019

Künstliche Intelligenz schlägt mehrfachen Weltmeister im Einsetzschach

Der von den TU-Studierenden Johannes Czech, Moritz Willig und Alena Beyer entwickelte Bot „CrazyAra" hat den Schachprofi Justin Tan in einem Online-Match der Schach-Variante „Crazyhouse" mit 4:1 geschlagen. Gelernt hat der Bot mittels künstlicher neuronaler Netze, was ihm erlaubt, vorausschauend Entscheidungen zu treffen. Das Besondere: Die Studierenden konnten damit einen Erfolg auf einem Feld feiern, das sonst von Giganten wie Google dominiert wird.

AI assists you



Your turn!

What do you think? Are we done? Is a AI just a success?

You have 5 minutes!

The New York Times

Opinion

A.I. Is Harder Than You Think

By Gary Marcus and Ernest Davis

Mr. Marcus is a professor of psychology and neural science. Mr. Davis is a professor of computer science.

May 18, 2018

AI has many isolated talents

AI is not superhuman

DARPA challenge (2015)

AI is not superhuman

And this also holds as of today

Your turn!

Do you think AI is superhuman? Please give examples and pros and cons. Also recall the definition of AI!

You have 5 minutes!

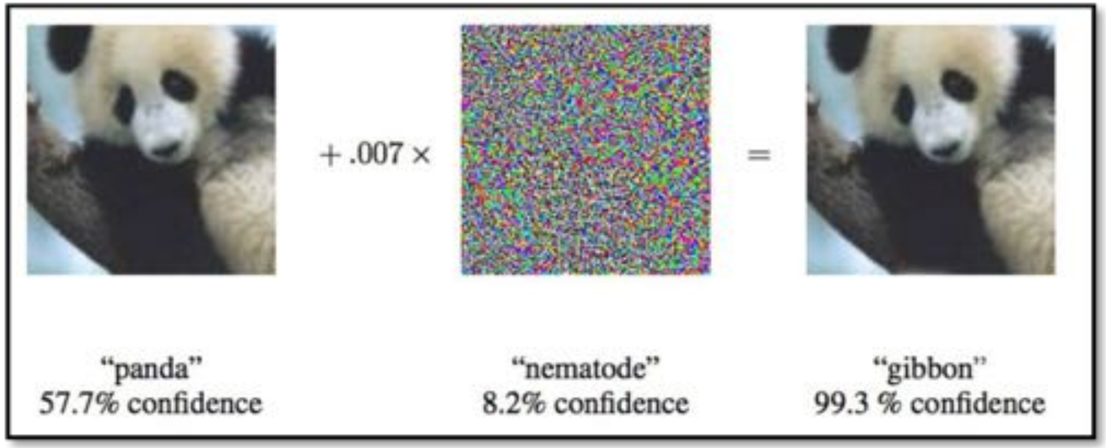
Fundamental Differences

as of today

Fundamental Differences

Sharif et al., 2015

Brown et al. (2017)



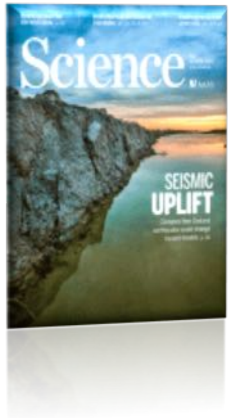
Google, 2015

REPORTS | PSYCHOLOGY

Semantics derived automatically from language corpora contain human-like biases

Aylin Caliskan[1,*], Joanna J. Bryson[1,2,*], Arvind Narayanan[1,*]
+ See all authors and affiliations

Science 14 Apr 2017:
Vol. 356, Issue 6334, pp. 183-186
DOI: 10.1126/science.aal4230



The Quest for a „good“ AI

How could an AI programmed by humans, with no more moral expertise than us, recognize (at least some of) our own civilization’s ethics as moral progress as opposed to mere moral instability?

„The Ethics of Artificial Intelligence“ Cambridge Handbook of Artificial Intelligence, 2011

Nick Bostrom

Future of Humanity Institute

UNIVERSITY OF OXFORD

Eliezer Yudkowsky

MIRI

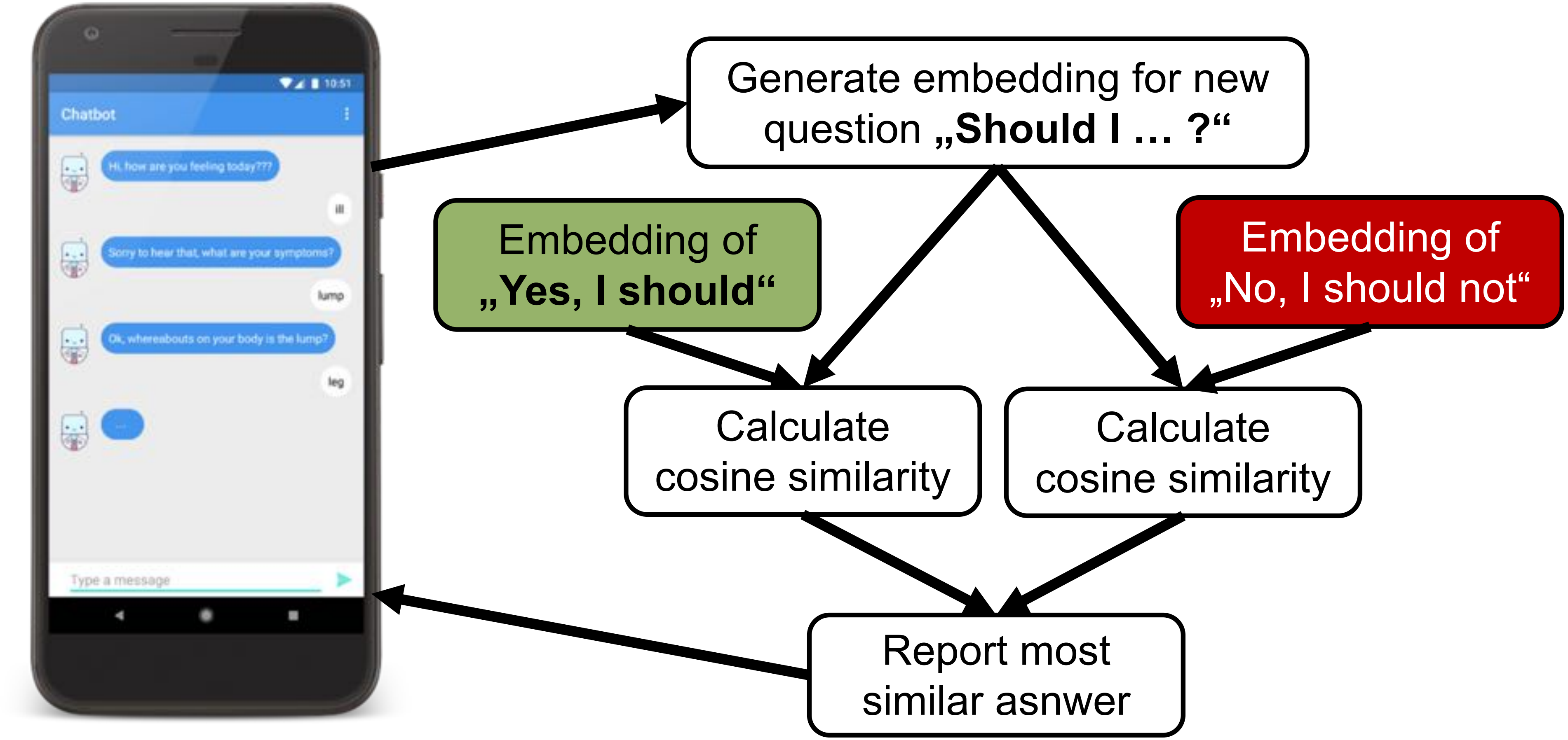
MACHINE INTELLIGENCE RESEARCH INSTITUTE

The Moral Choice Machine

Not all stereotypes are bad

[Jentzsch, Schramowski, Rothkopf, Kersting AIES 2019]

AAAI / ACM conference on ARTIFICIAL INTELLIGENCE, ETHICS, AND SOCIETY



The Moral Choice Machine

Not all stereotypes are bad

[Jentzsch, Schramowski, Rothkopf, Kersting AIES 2019]

AAAI / ACM conference on
ARTIFICIAL INTELLIGENCE, ETHICS, AND SOCIETY

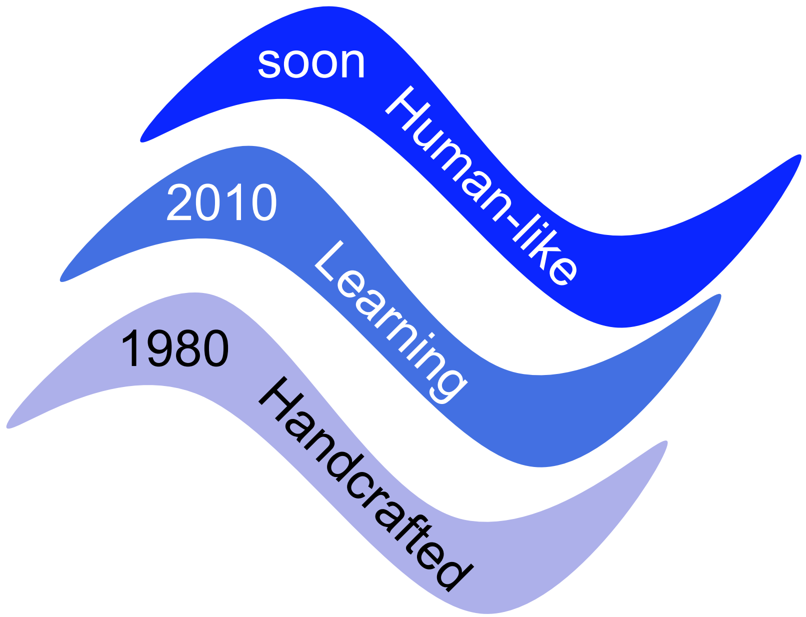
TECHNISCHE UNIVERSITÄT DARMSTADT

https://www.hr-fernsehen.de/sendungen-a-z/hauptsache-kultur/sendungen/hauptsache-kultur,sendung-56324.html

Video 05:10 Min.
Der Hamster gehört nicht in den Toaster – Wie Forscher von der TU Darmstadt versuchen, Maschinen ... [Videoseite]
hauptsache kultur | 14.03.19, 22:45 Uhr

The future of AI

The future of AI
The third wave of AI



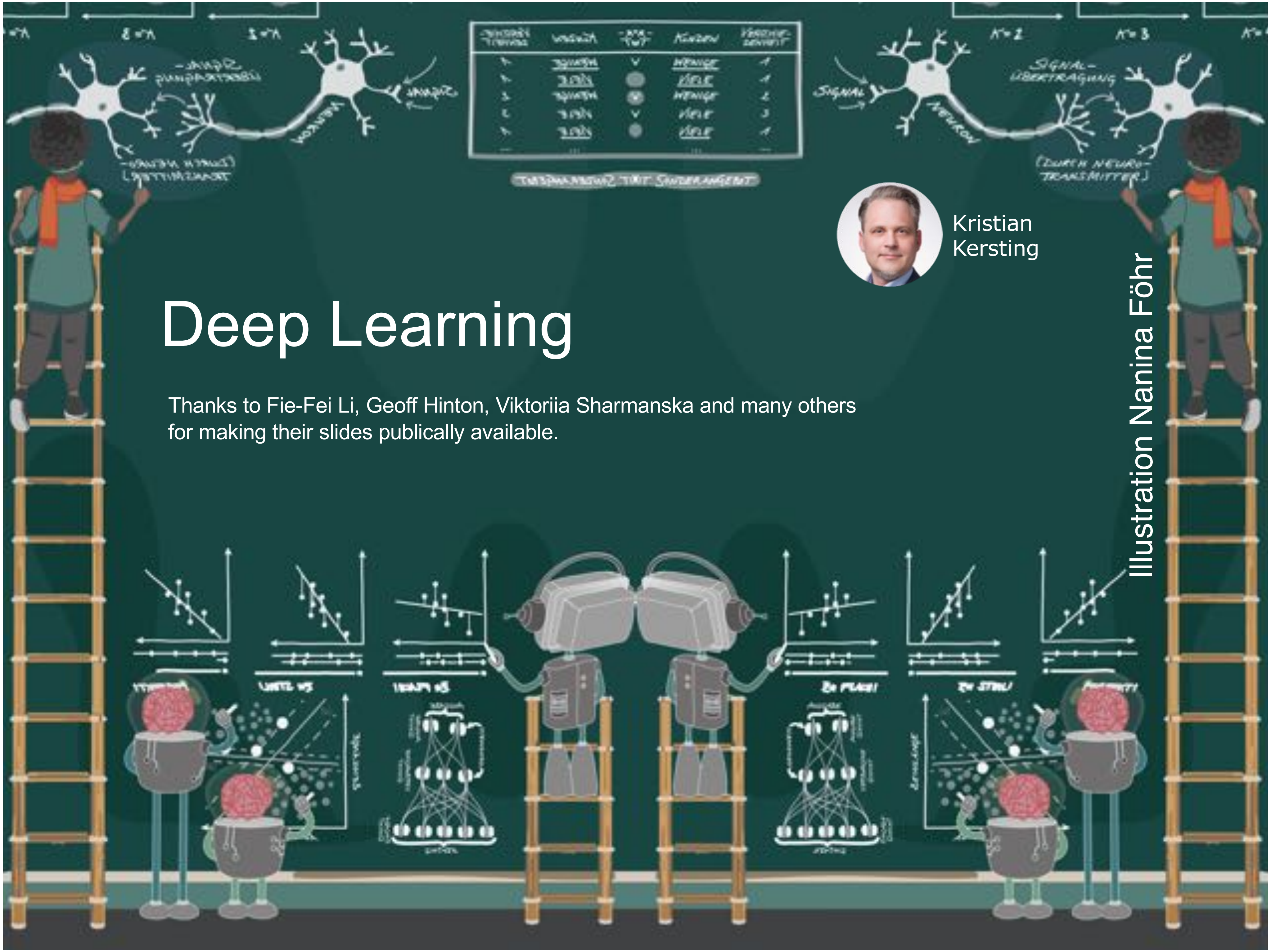
AI systems that can acquire human-like communication and reasoning capabilities, with the ability to recognise new situations and adapt to them.

Meeting this grand challenge is a team sport !

And this is AI! Still a lot to be done! It is a team sport.

Thanks to all students of the Research Training Group "Artificial Intelligence - Facts, Chances, Risks“ of the German National Academic Scholarship Foundation. Special thanks to **Maike Elisa Müller** and **Jannik Kossen** for taking the lead and to **Matthias Kleiner**, president of the Leibniz Association, for his preface

Illustration Nanina Föhr



Kristian Kersting

Deep Learning

Thanks to Fie-Fei Li, Geoff Hinton, Viktoriia Sharmanska and many others for making their slides publically available.

Illustration Nanina Föhr

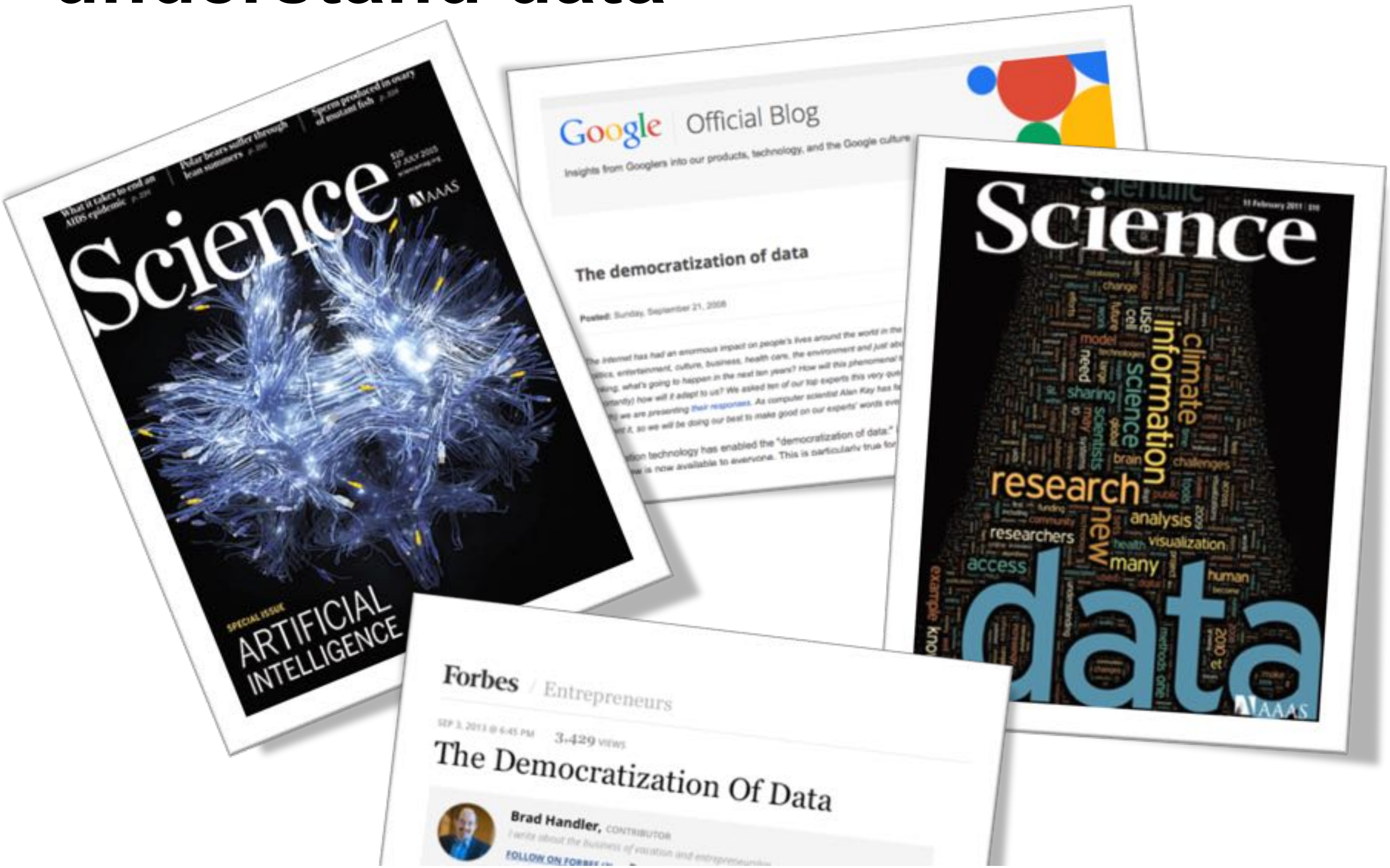
Your turn!

So we know what algorithms are! Are they just for computers? What do you think?

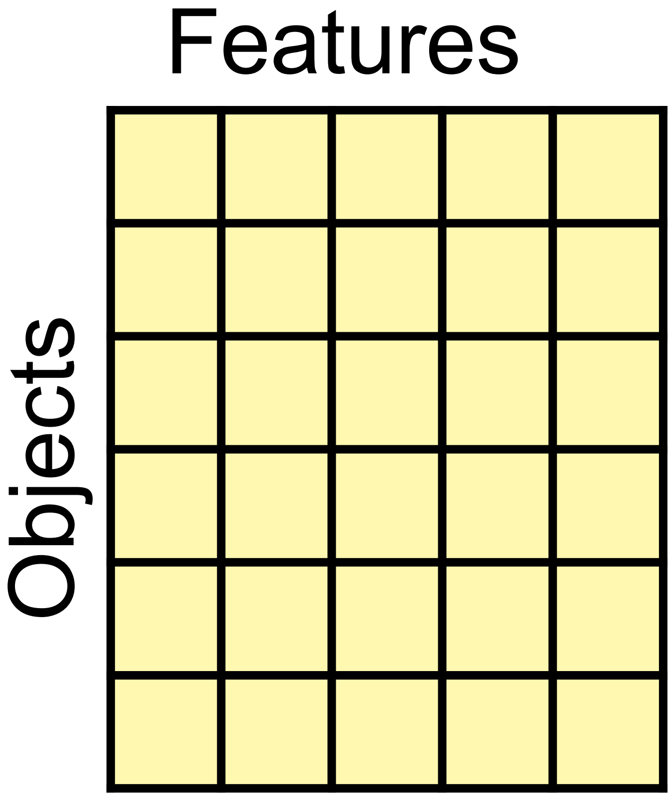
You have 5 minutes!

Algorithms are not just for computers

Arms race to deeply understand data

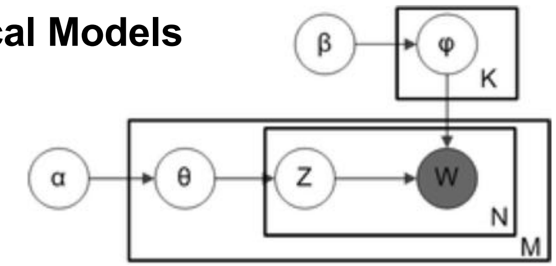


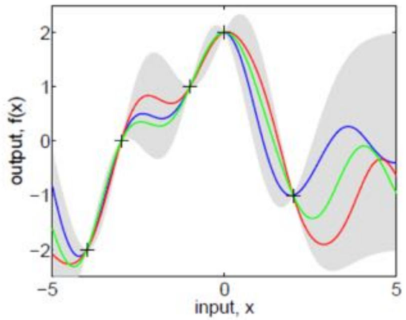
Bottom line:
Take your data spreadsheet …



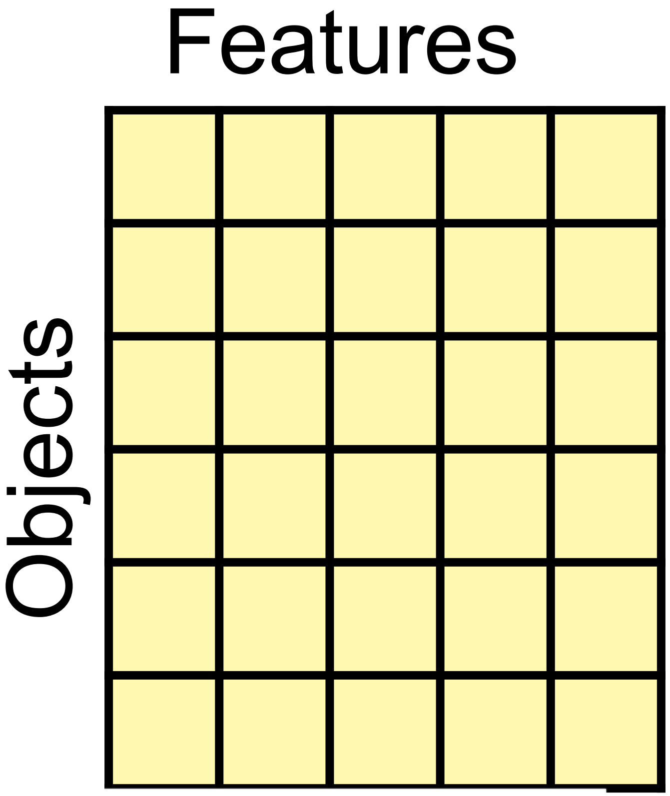
… and apply Machine Learning

Probabilistic Graphical Models
Arithmetic Circuits

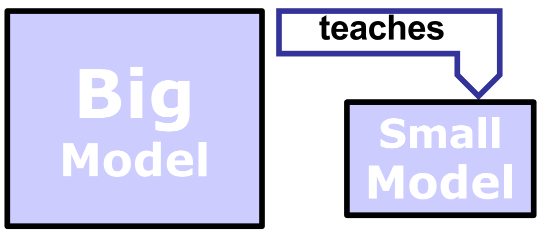




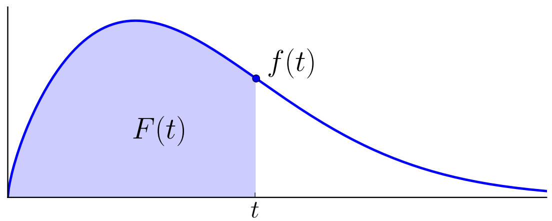
Gaussian Processes



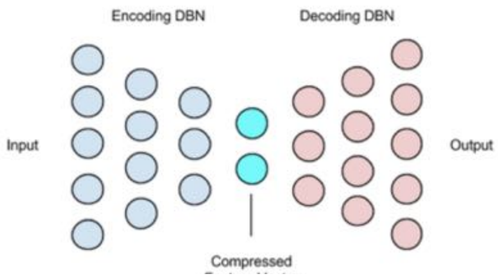
Boosting



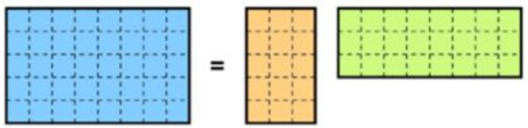
Distillation/LUPI



Diffusion Models



Autoencoder,
Deep Learning



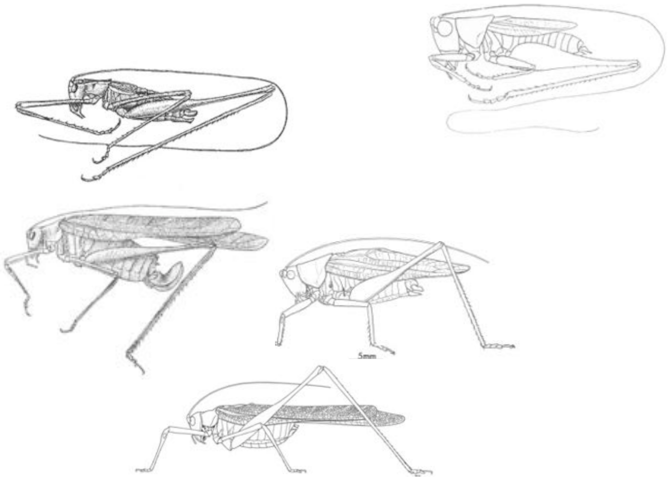
Big Data Matrix Factorization

and many more …

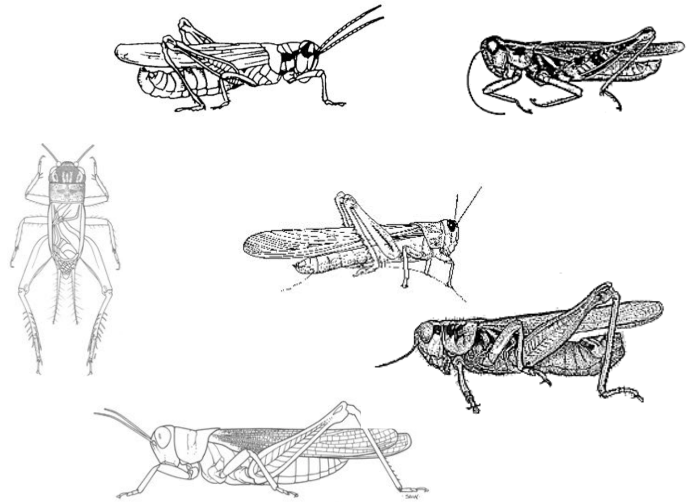
We have 10 example.

5 “**Laubheuschrecken**“ and 5 **Grashüpfer.**

Laubheuschrecken



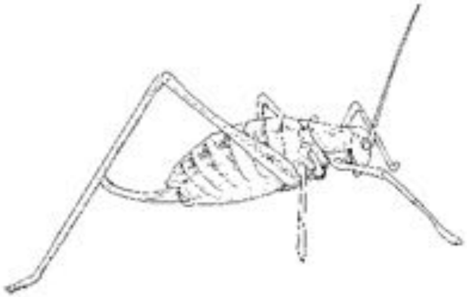
Grashüpfer



Let us put the examples into an Excel sheet

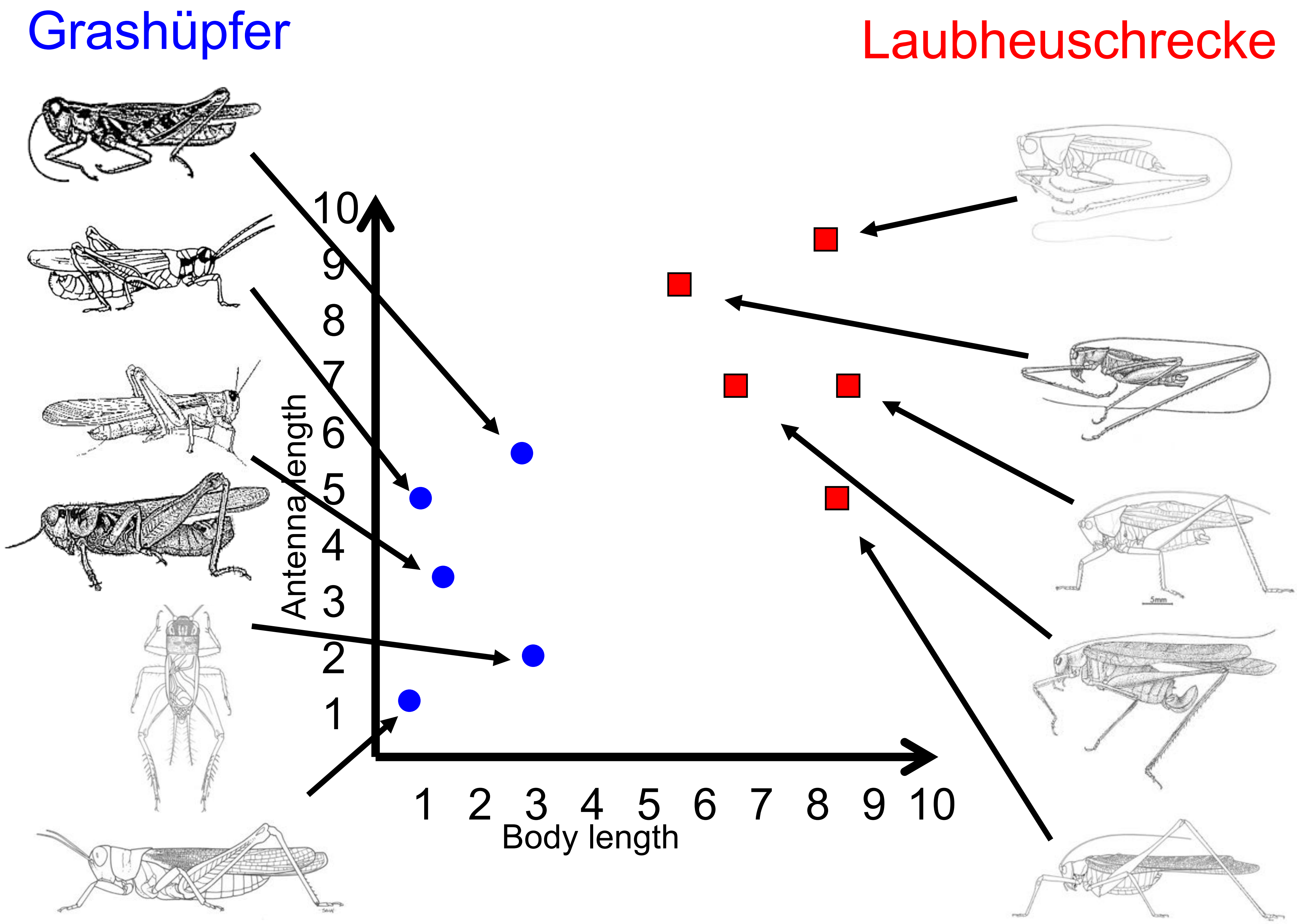
Not a feature, just for organization!!!!

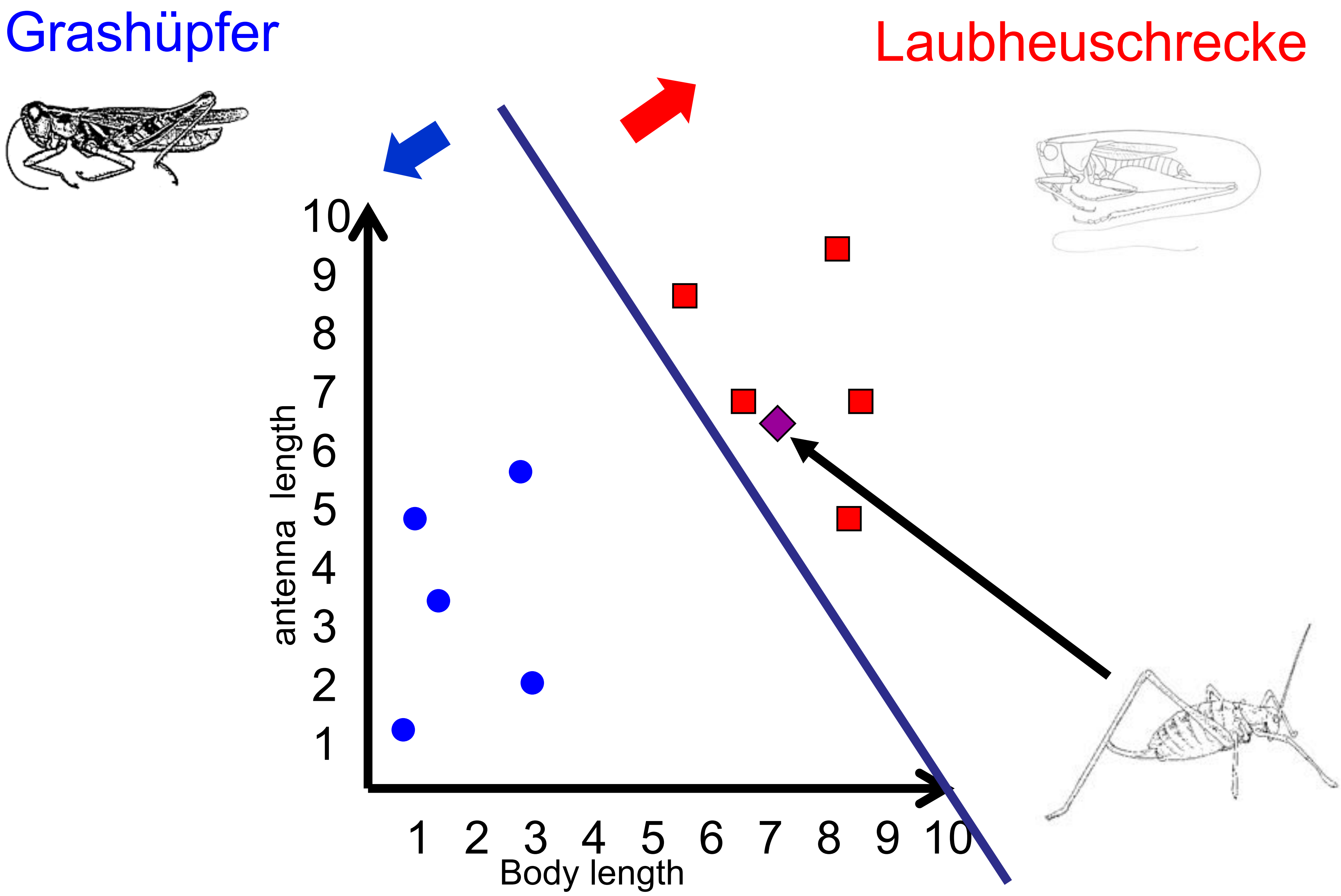
ID	Body length	antenna length	Class
1	2.7	5.5	Grasshüpfer
2	8.0	9.1	Laubheuschrecke
3	0.9	4.7	Grasshüpfer
4	1.1	3.1	Grasshüpfer
5	5.4	8.5	Laubheuschrecke
6	2.9	1.9	Grasshüpfer
7	6.1	6.6	Laubheuschrecke
8	0.5	1.0	Grasshüpfer
9	8.3	6.6	Laubheuschrecke
10	8.1	4.7	Laubheuschrecke



11	5.1	7.0	?

Laubheuschrecke or **Grasshüpfer**?

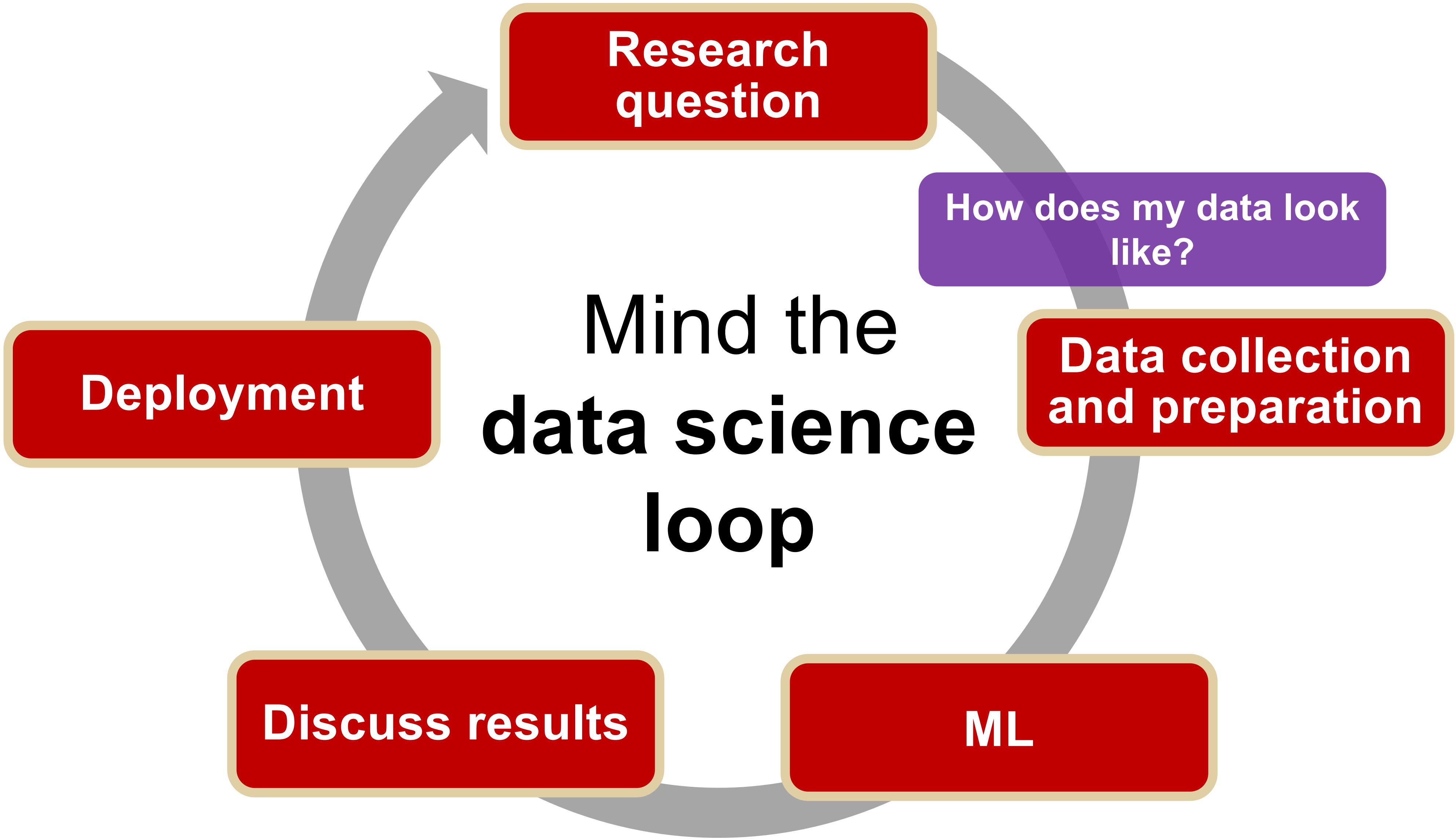

Grashüpfer
Laubheuschrecke
10
9
8
7
6
5
4
3
2
1
Antenna length
1 2 3 4 5 6 7 8 9 10
Body length
5mm


Grashüpfer
Laubheuschrecke
10
9
8
7
6
5
4
3
2
1
antenna length
1 2 3 4 5 6 7 8 9 10
Body length

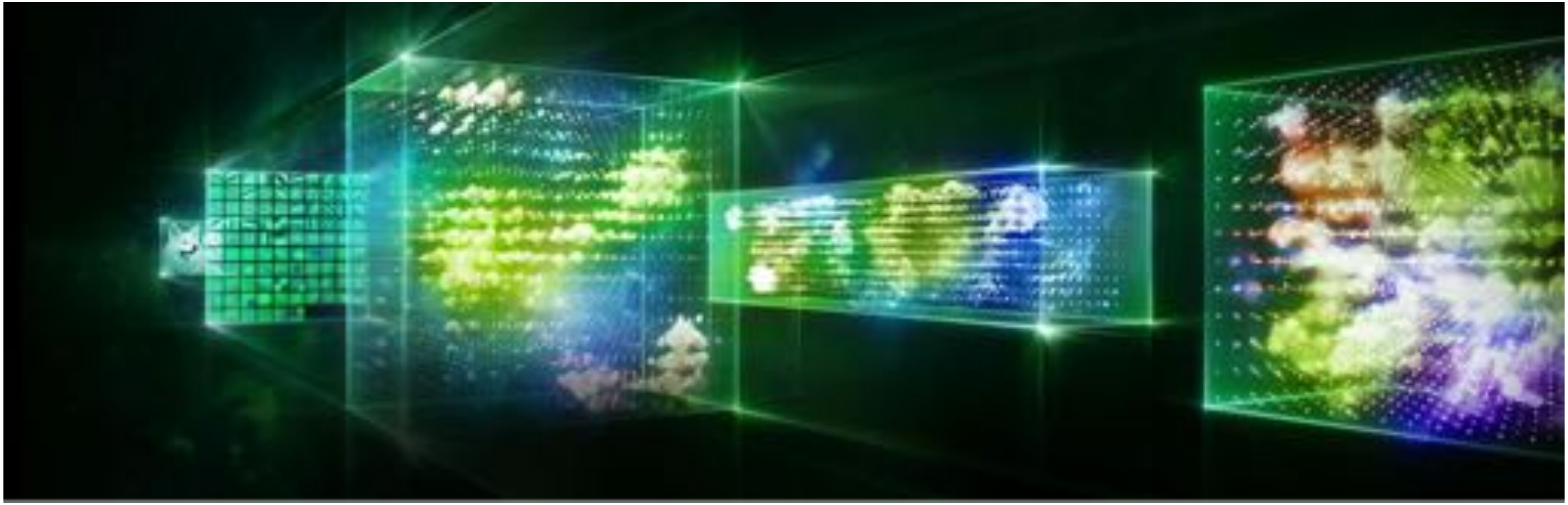
Your turn!

Simple! What do you think? Is machine learning that simple?

You have 5 minutes!

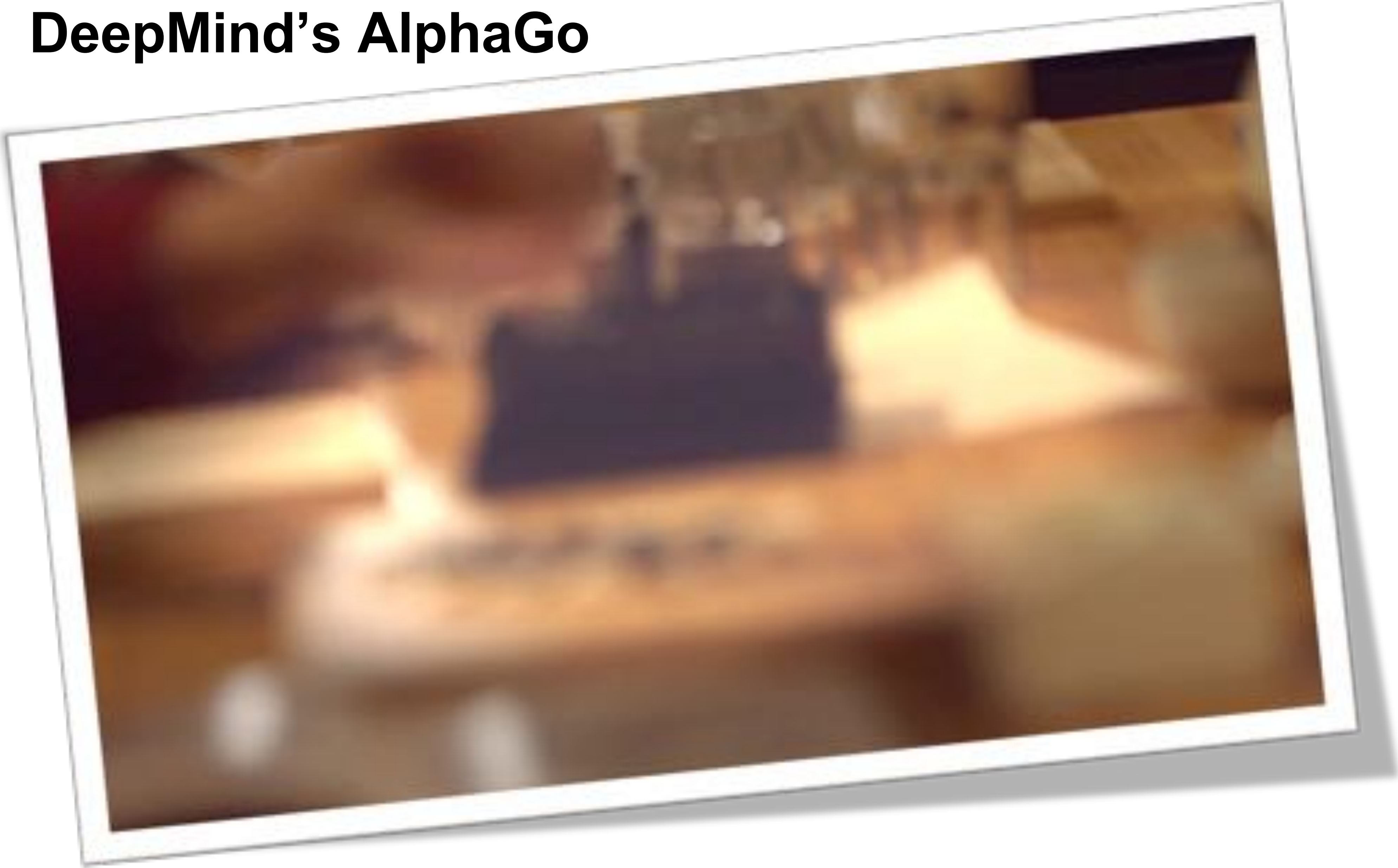

Research question
How does my data look like?
Data collection and preparation
ML
Discuss results
Deployment
Mind the data science loop

What if the machine can help to find the right representation?



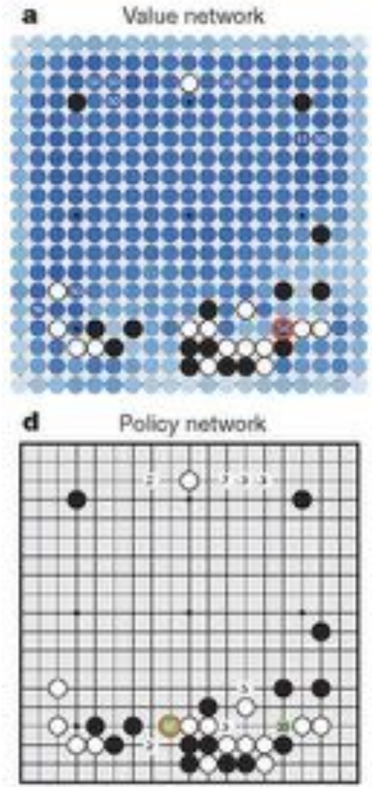
Deep Neural Learning

DeepMind's AlphaGo



Watch NATURE video at https://www.youtube.com/watch?v=g-dKXOIsf98

DeepMind's AlphaGo



Deep policy network is trained to produce probability map of promising moves. The deep value network is used to prune the search tree (monte-carlo tree search); so there is a lot of classical AI machinery around the deep (p)art.

And yes, the machine may also learn to play other games

Goal of Deep Architectures

High-level semenatical representations

To this aim most approaches use (stacked) neural networks

Edges, local shapes, object parts

Low level representation

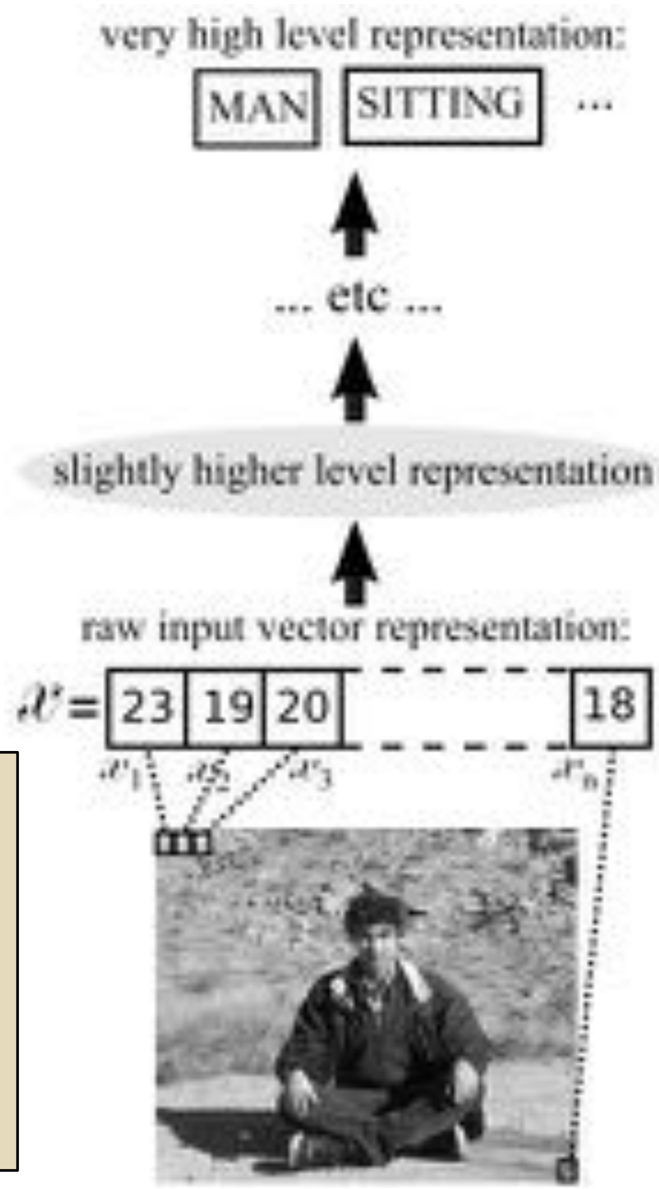


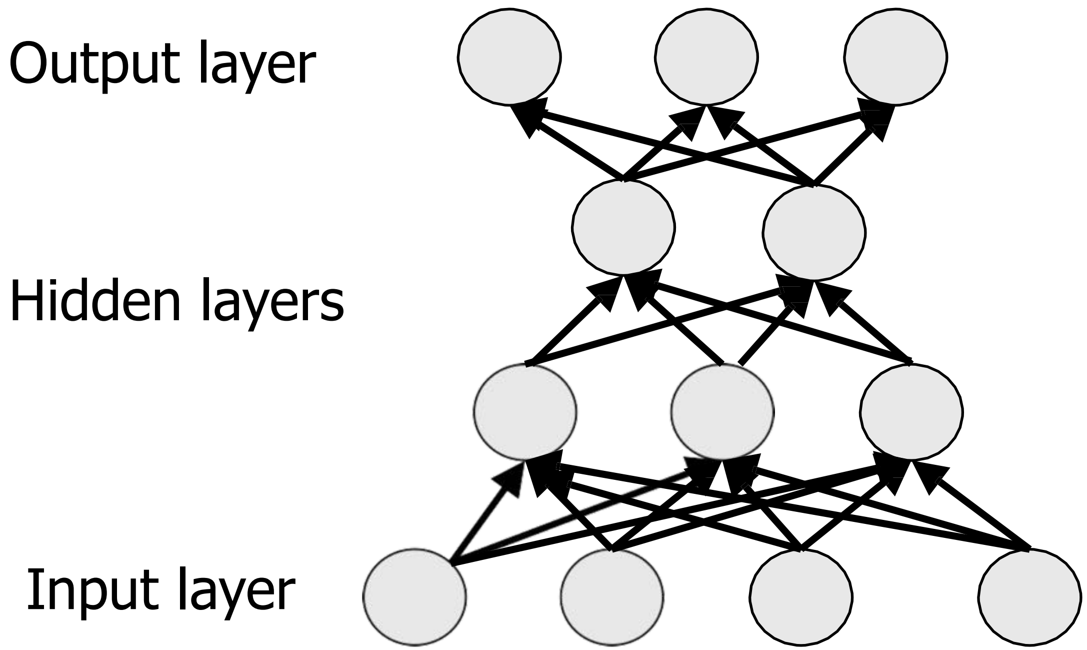
Figure is from Yoshua Bengio

Deep learning methods aim at

- **learning feature hierarchies**
- where features from higher levels of the hierarchy are formed by lower level features.

Deep Architectures

Deep architectures are composed of multiple levels of non-linear operations, such as neural nets with many hidden layers.



Examples of non-linear activations:

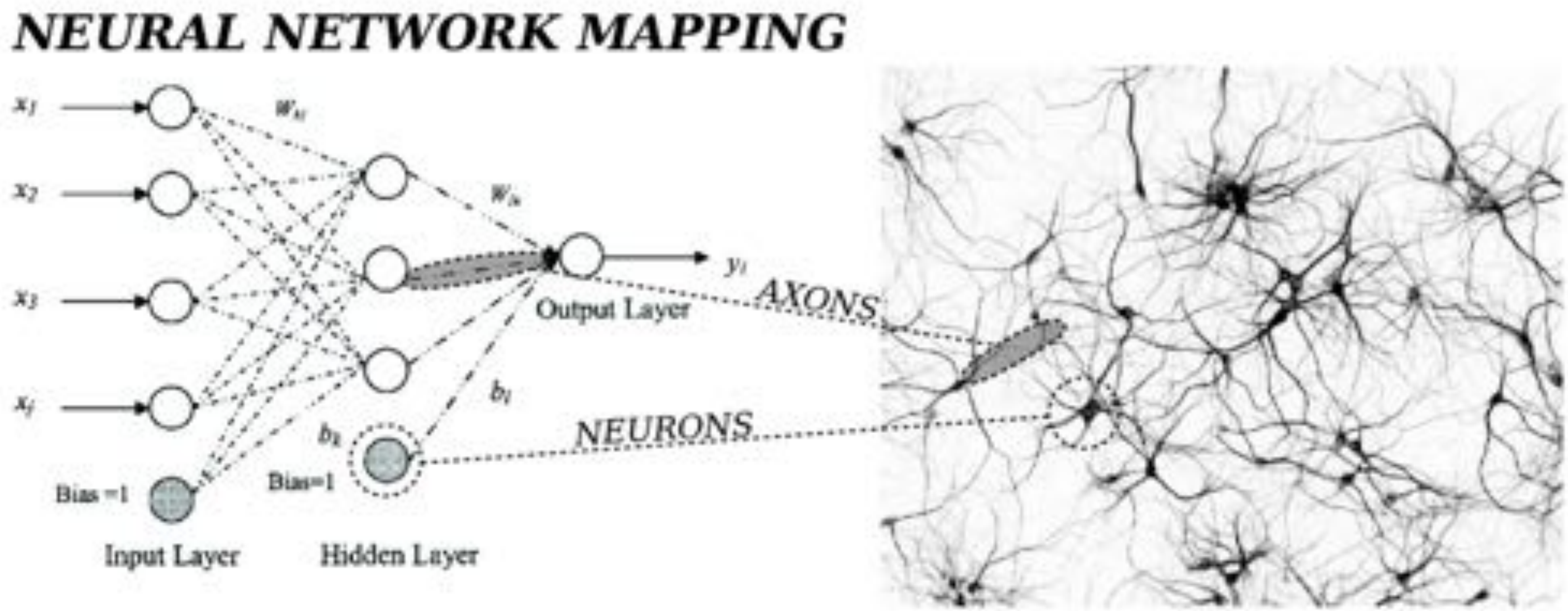
$\tanh(x)$

$\sigma(x) = (1 + e^{-x})^{-1}$

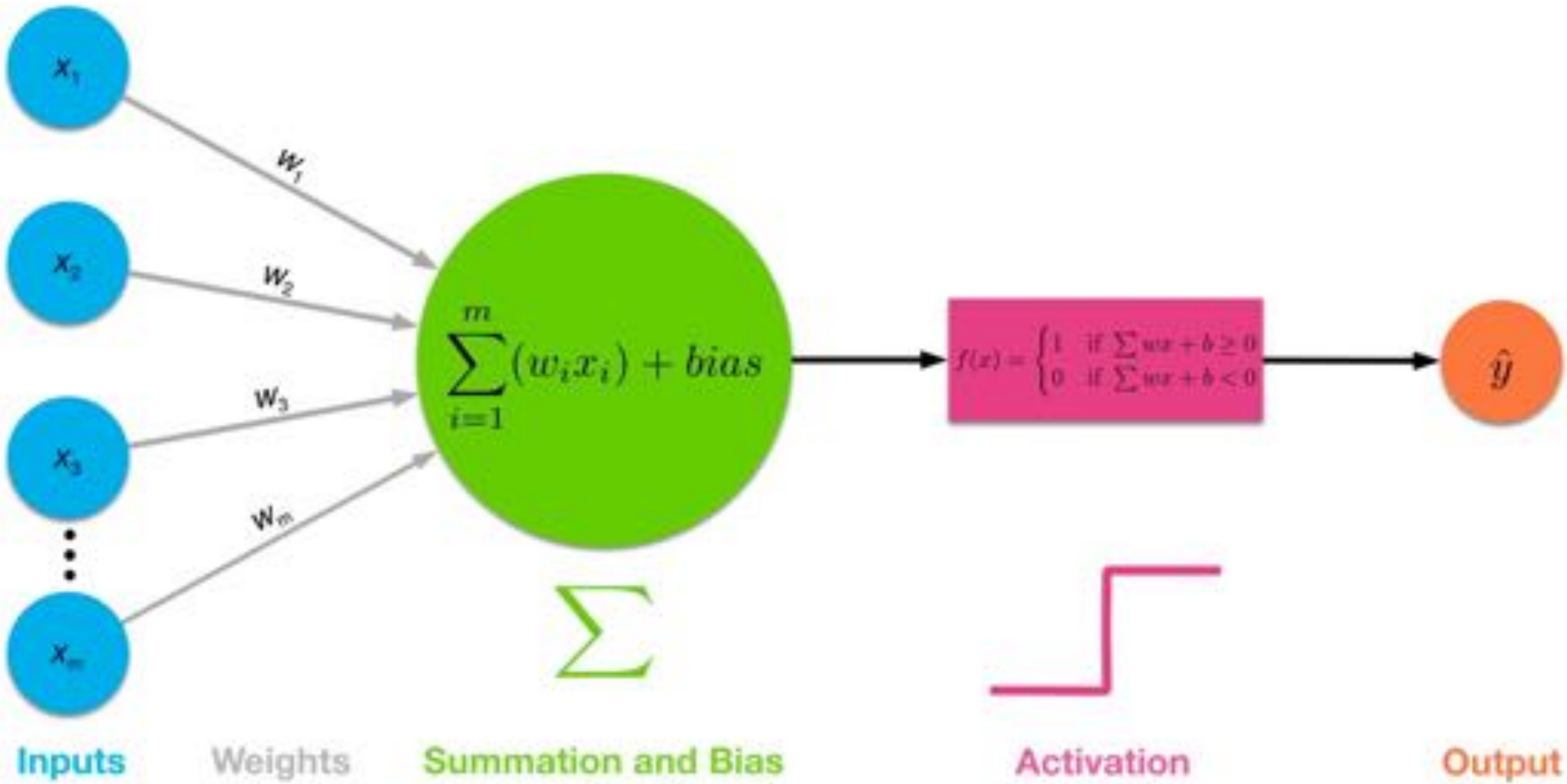
$\max(0, x)$

In practice, NN with multiple hidden layers work better than with a single hidden layer.

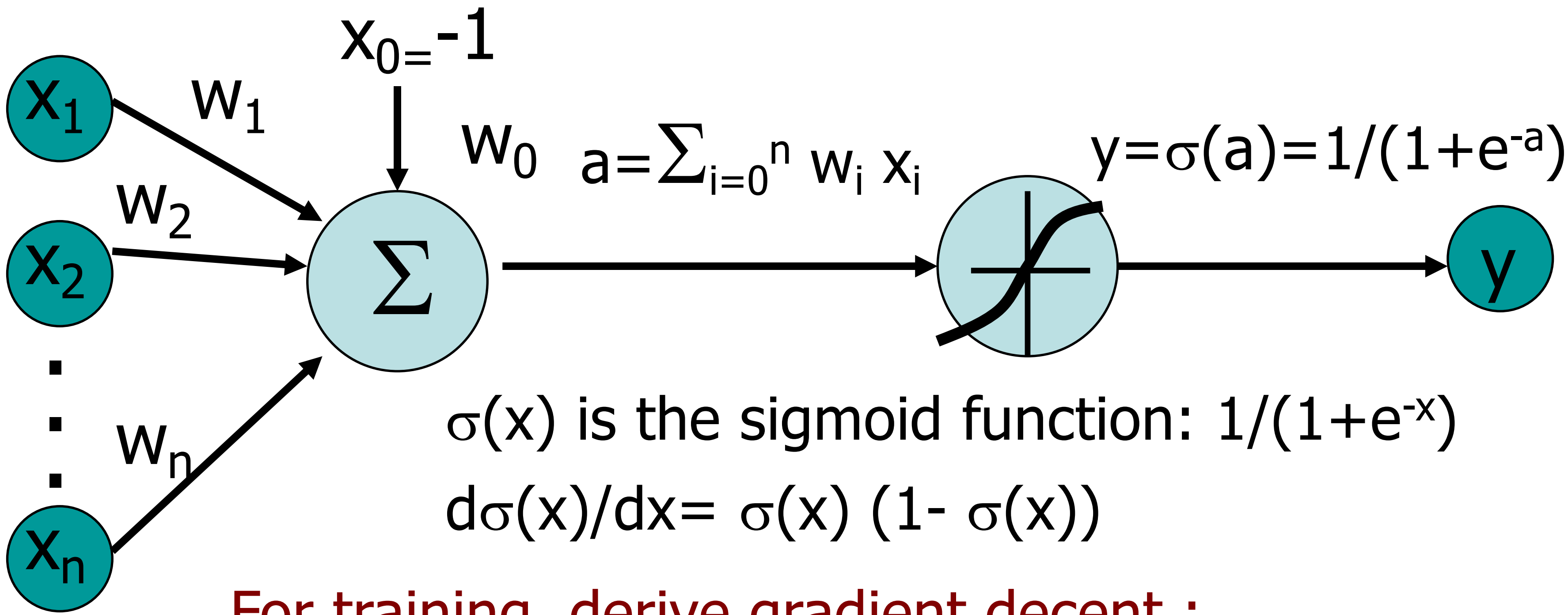
Artificial Neural Networks are inspired by neural networks



Abstract Neural Unit



Commonly, neurons are encoded as **Sigmoid Unit (but other units are possible)**



$\sigma(x)$ is the sigmoid function: $1/(1+e^{-x})$

$d\sigma(x)/dx= \sigma(x)\ (1- \sigma(x))$

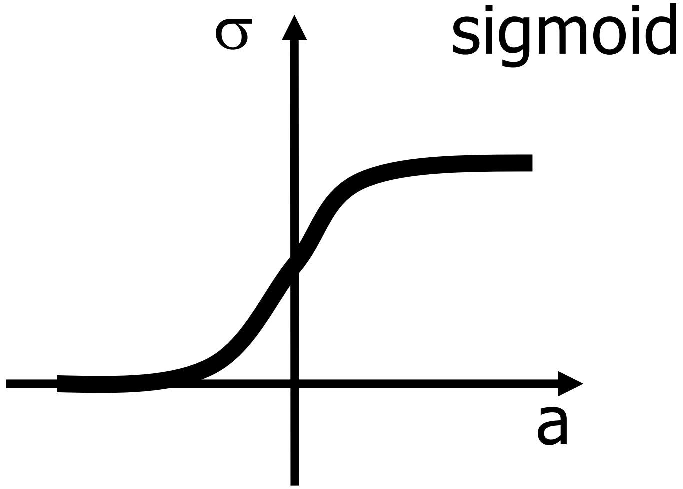
For training, derive gradient decent :

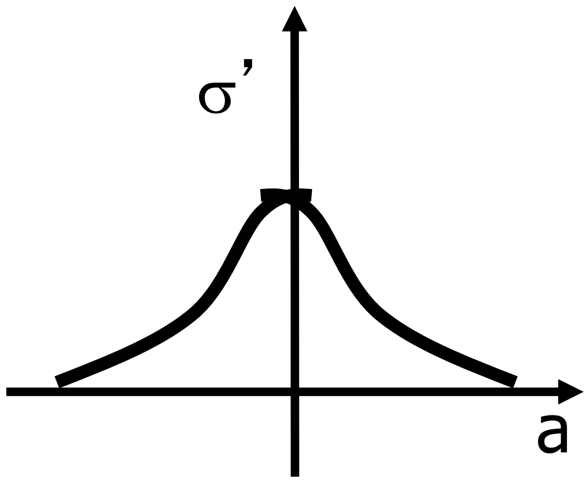
- one sigmoid function

  $\partial E/\partial w_i = -\sum_p (t^p-y^p)\ y^p\ (1-y^p)\ x_i^p$

- Multilayer networks of sigmoid units use **backpropagation**

Gradient Descent Rule for Sigmoid Output Function





$$E^p[w_1,\ldots,w_n] = \tfrac{1}{2}\,(t^p-y^p)^2$$

$$\partial E^p/\partial w_i = \partial/\partial w_i \ \tfrac{1}{2}\,(t^p-y^p)^2$$

$$= \partial/\partial w_i \ \tfrac{1}{2}\,(t^p-\sigma(\textstyle\sum_i w_i\, x_i^p))^2$$

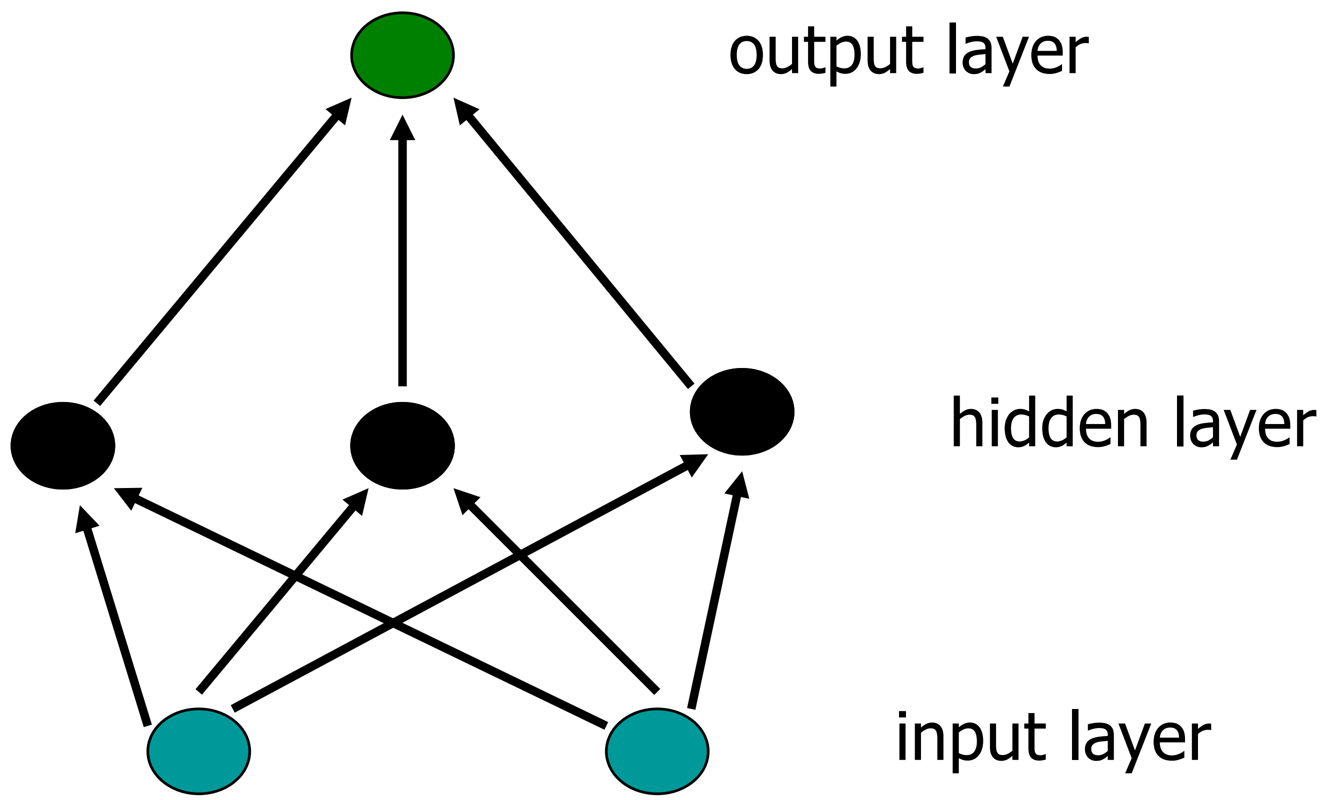
$$= (t^p-y^p)\ \underline{\sigma'(\textstyle\sum_i w_i\, x_i^p)}\ (-x_i^p)$$

for $y=\sigma(a) = 1/(1+e^{-a})$

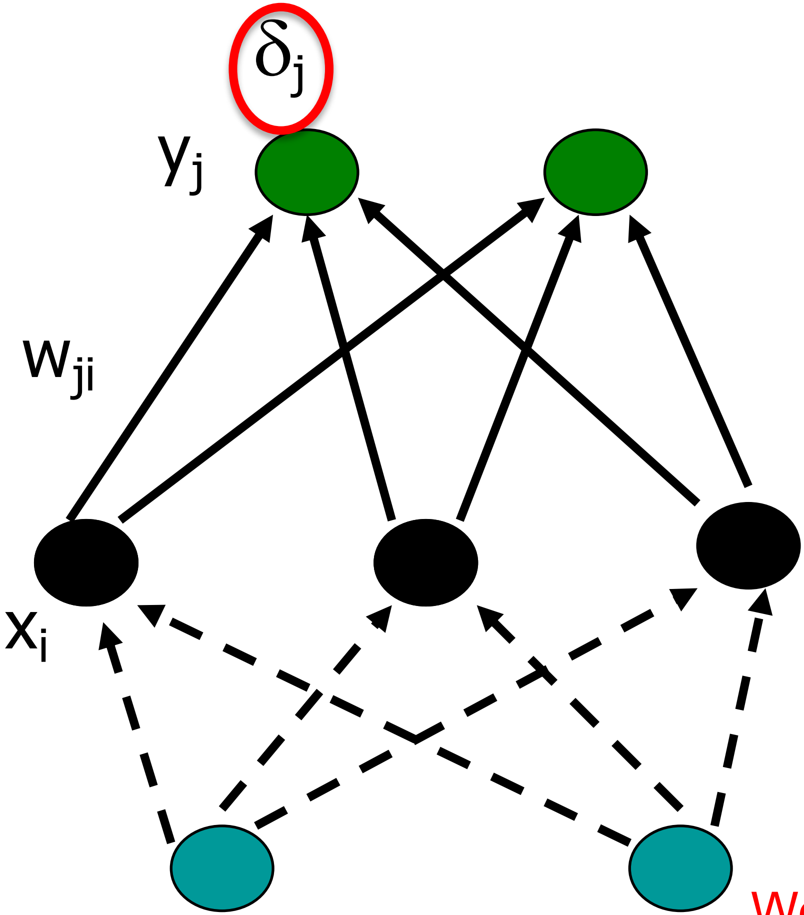
$\sigma'(a)= e^{-a}/(1+e^{-a})^2=\underline{\sigma(a)\ (1-\sigma(a))}$

$$w'_i= w_i + \alpha\ \overline{y^p\,(1-y^p)}(t^p-y^p)\ x_i^p$$

Build (feedforward) Multi-Layer Networks by sticking together units



Training-Rule for Weights to the Output Layer



$$E^p[w_{ij}] = ½ \Sigma_j (t_j^p - y_j^p)^2$$

$$\partial E^p/\partial w_{ji} = \partial/\partial w_{ji}\ ½ \Sigma_j (t_j^p - y_j^p)^2$$
$$= \ldots$$
$$= - y_j^p(1-y^p_j)(t^p_j-y^p_j)\ \ x_i^p$$

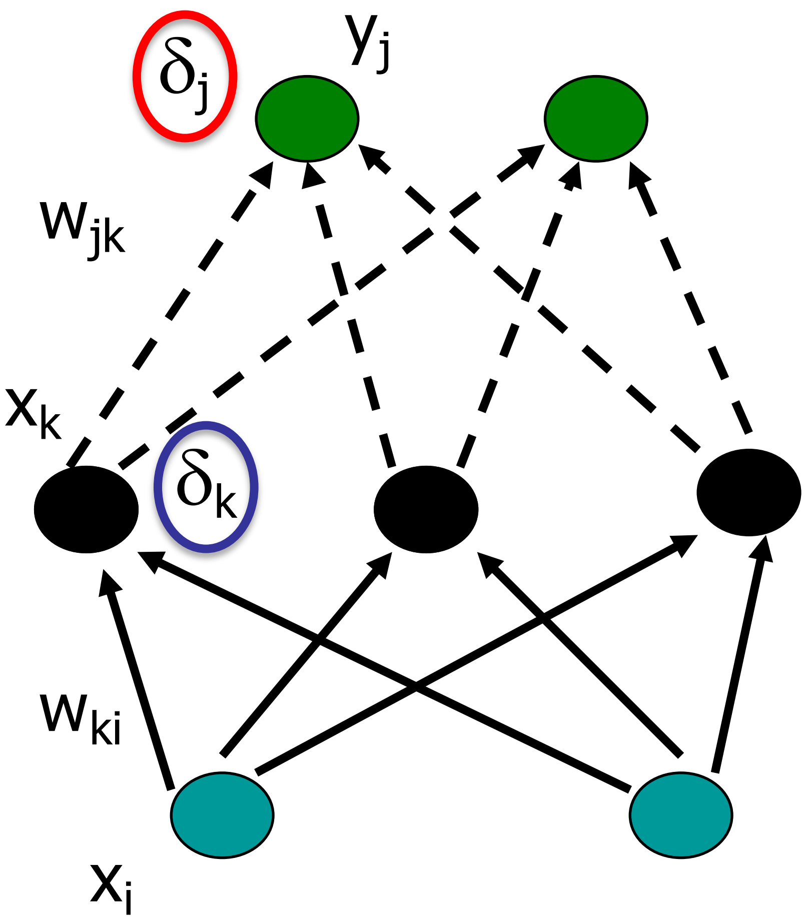
$$\Delta w_{ji} = \alpha\ \ y_j^p(1-y_j^p)\ (t^p_j-y_j^p)\ x_i^p$$
$$= \alpha\ \ \delta_j^p\ x_i^p$$

activation

We just want to rewrite in terms of input-output only

with $\delta_j^p := y_j^p(1-y_j^p)\ (t^p_j-y_j^p)$

Training-Rule for Weights to the Output Layer



Credit assignment problem:
No target values t for hidden layer units.

Error for hidden units?

$$\delta_k = \Sigma_j \, w_{jk} \, \delta_j \, y_j \, (1-y_j)$$

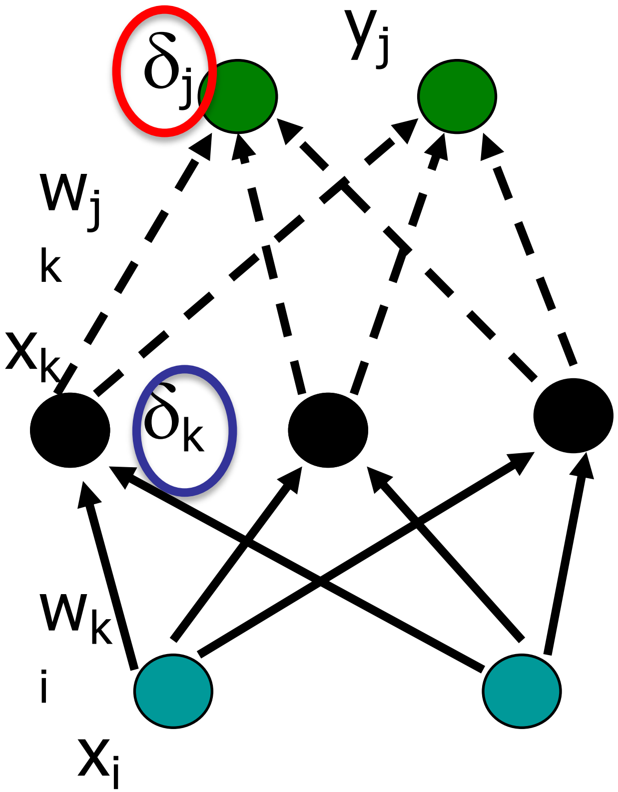
$$\Delta w_{ki} = \alpha \; x_k^p(1-x_k^p) \, \delta_k^p \, x_i^p$$

activation

View x_k as intermediate output

activation

Training-Rule for Weights to the Output Layer

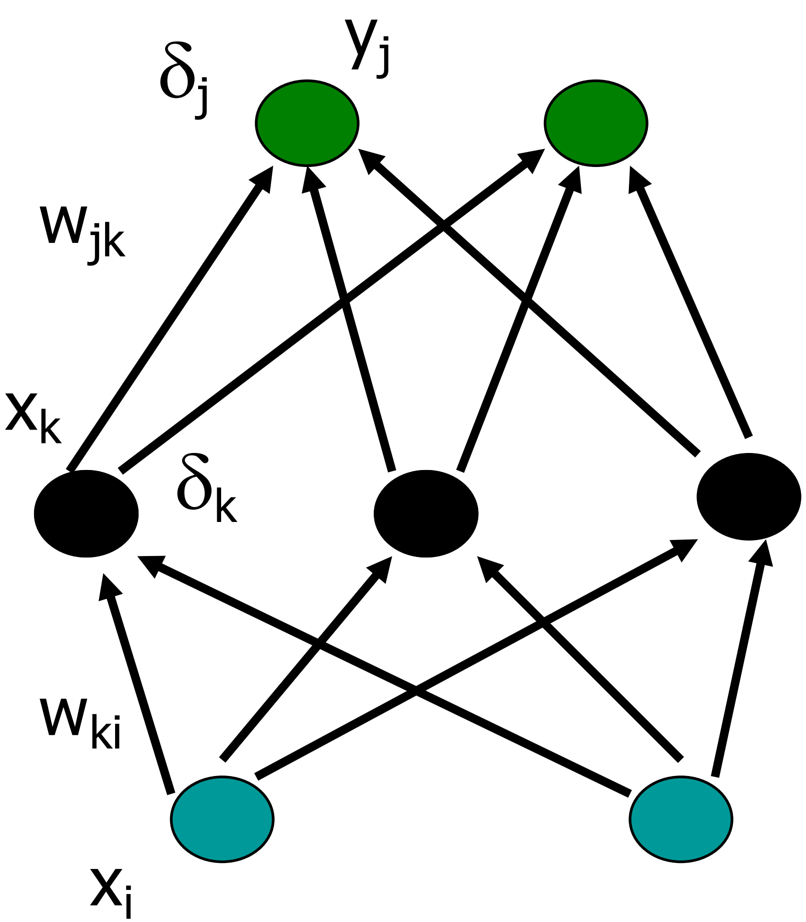


$$E^p[w_{ki}] = \tfrac{1}{2} \Sigma_j (t_j^p - y_j^p)^2$$

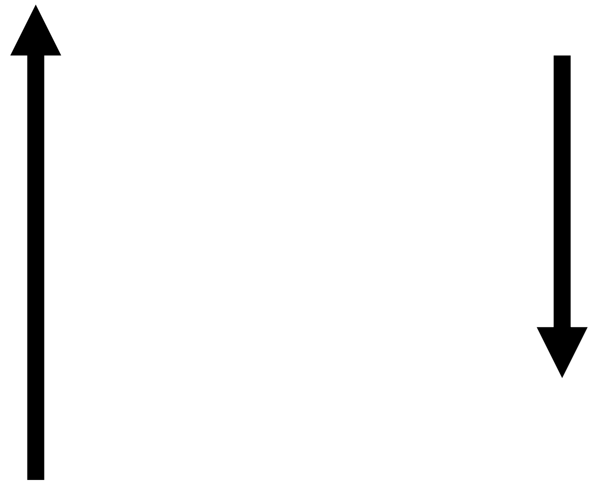
$$\partial E^p / \partial w_{ki} = \partial / \partial w_{ki} \tfrac{1}{2} \Sigma_j (t_j^p - y_j^p)^2$$
$$= \partial / \partial w_{ki} \tfrac{1}{2} \Sigma_j (t_j^p - \sigma(\Sigma_k w_{jk} x_k^p))^2$$
$$= \partial / \partial w_{ki} \tfrac{1}{2} \Sigma_j (t_j^p - \sigma(\Sigma_k w_{jk} \sigma(\Sigma_i w_{ki} x_i^p)))^2$$
$$= -\Sigma_j (t_j^p - y_j^p) \sigma'_j(a) w_{jk} \sigma'_k(a) x_i^p$$
$$= -\Sigma_j \delta_j w_{jk} \sigma'_k(a) x_i^p$$
$$= -\Sigma_j \delta_j w_{jk} x_k (1 - x_k) x_i^p$$

$$\Delta w_{ki} = \alpha \delta_k x_i^p \quad \text{with } \delta_k = \Sigma_j \delta_j w_{jk} x_k(1 - x_k)$$

Backpropagation

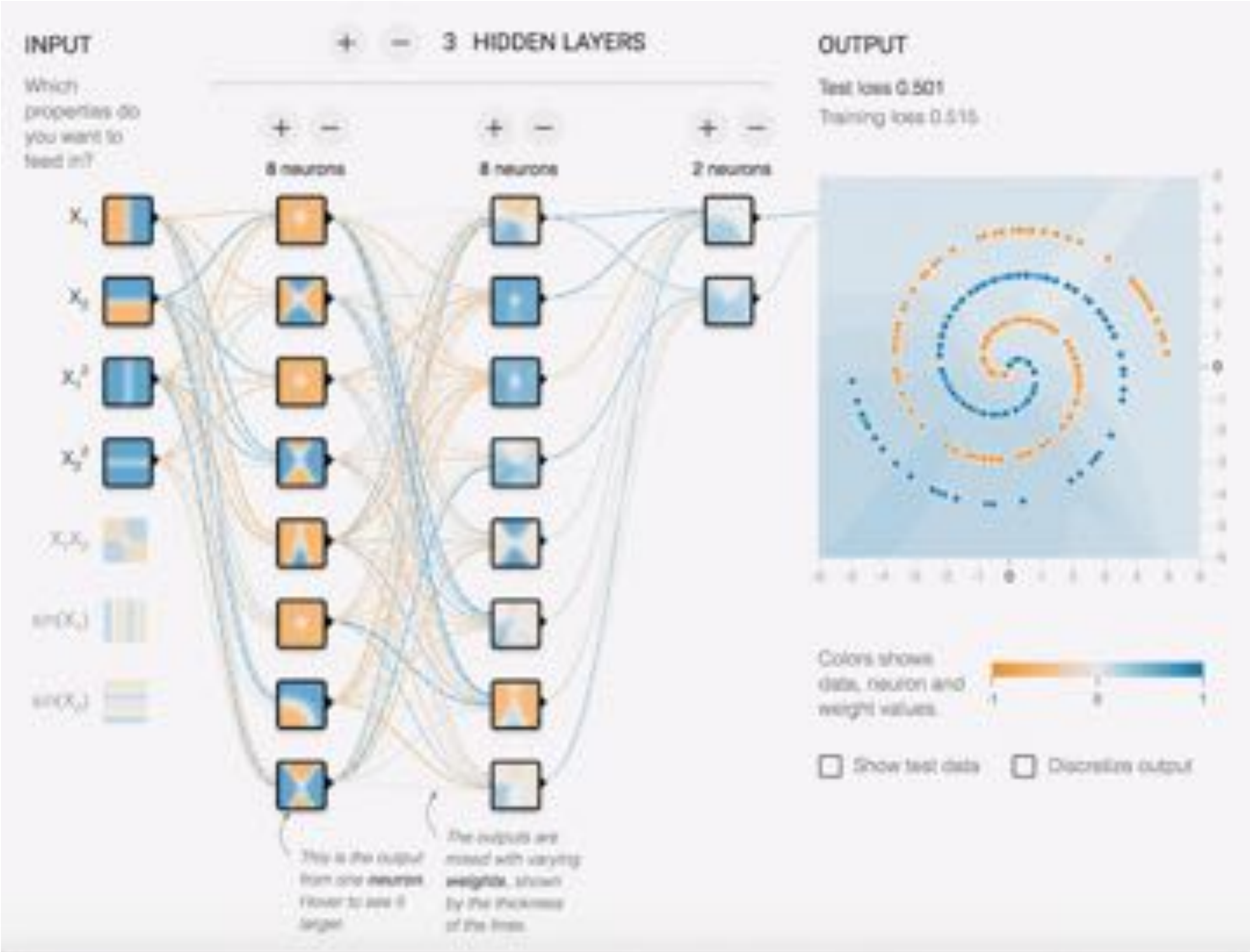


Backward step:
propagate errors from output to hidden layer



Forward step:
Propagate activation from input to output layer

Tinker with a neural network at http://playground.tensorflow.org/

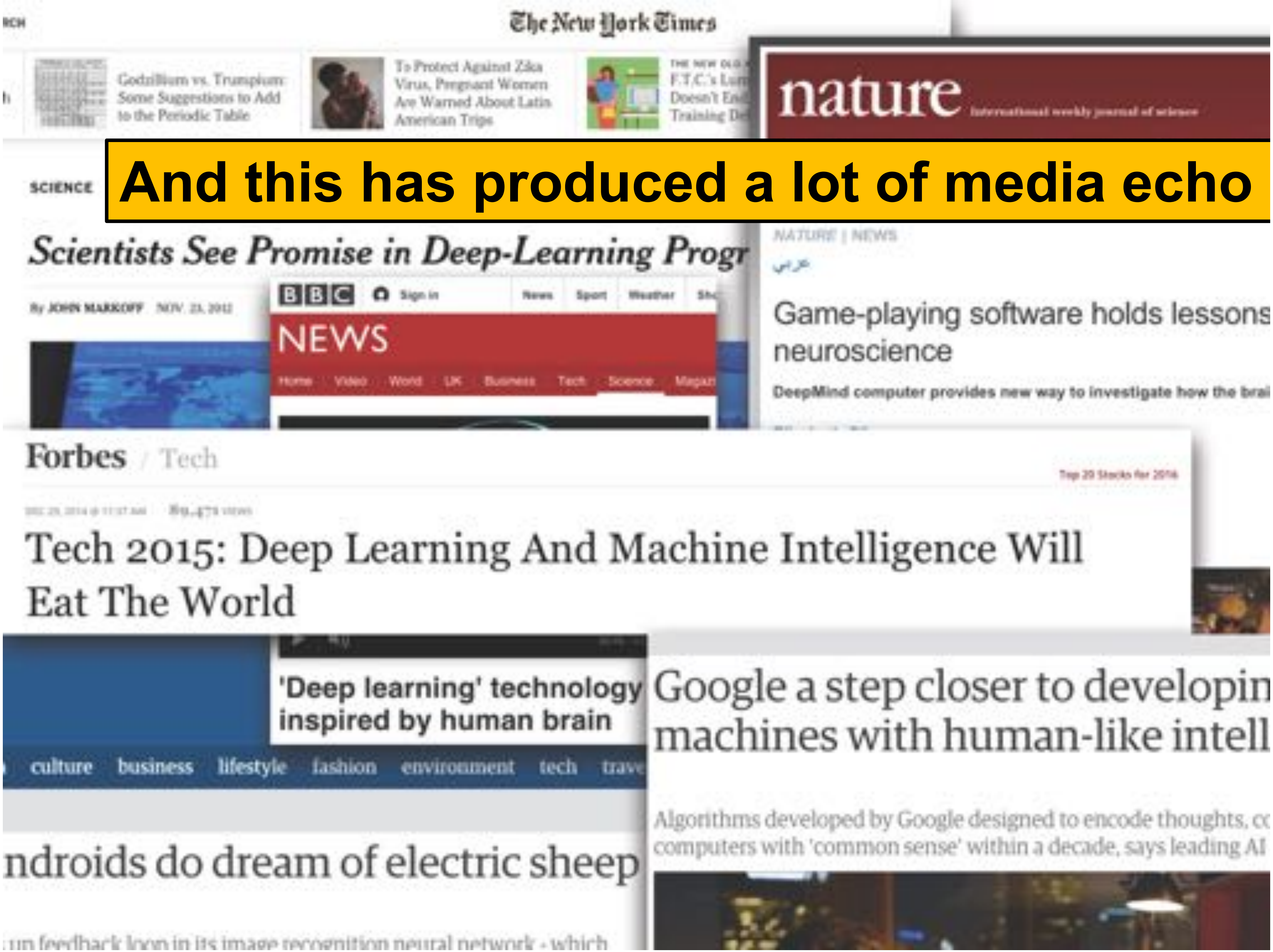


Your turn!

What do you think? Are artificial neural networks biologically plausible?

You have 5 minutes!

And this has produced a lot of media echo



The first breakthrough of (D)NNs was on image classification

Deep Convolutional Networks

- ❑ Convolutional layer
- ❑ Non-linear activation function ReLU
- ❑ Max pooling layer
- ❑ Fully connected layer

Deep Convolutional Networks CNNs

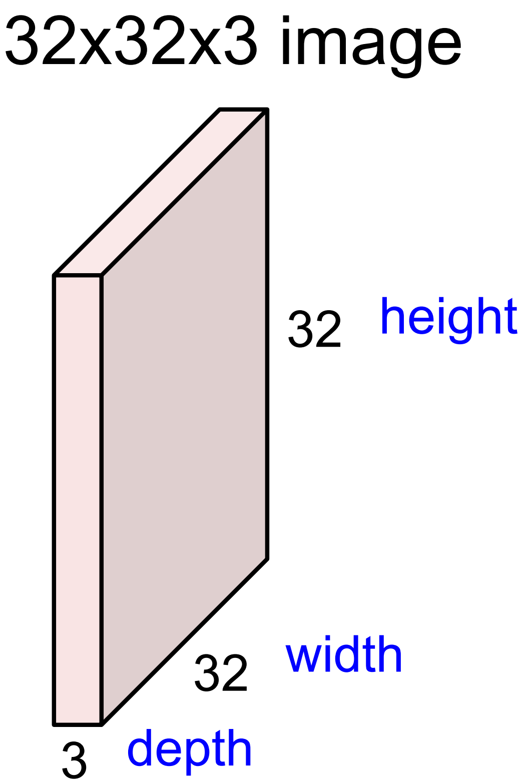
Compared to standard neural networks with similarly-sized layers,

- CNNs have much fewer connections and parameters
- and so they are easier to train
- and typically have more than five layers (a number of layers which makes fully-connected neural networks almost impossible to train properly when initialized randomly)
- and they are tailored towards computer vision

LeNet, 1998 LeCun Y, Bottou L, Bengio Y, Haffner P: Gradient-Based Learning Applied to Document Recognition, Proceedings of the IEEE

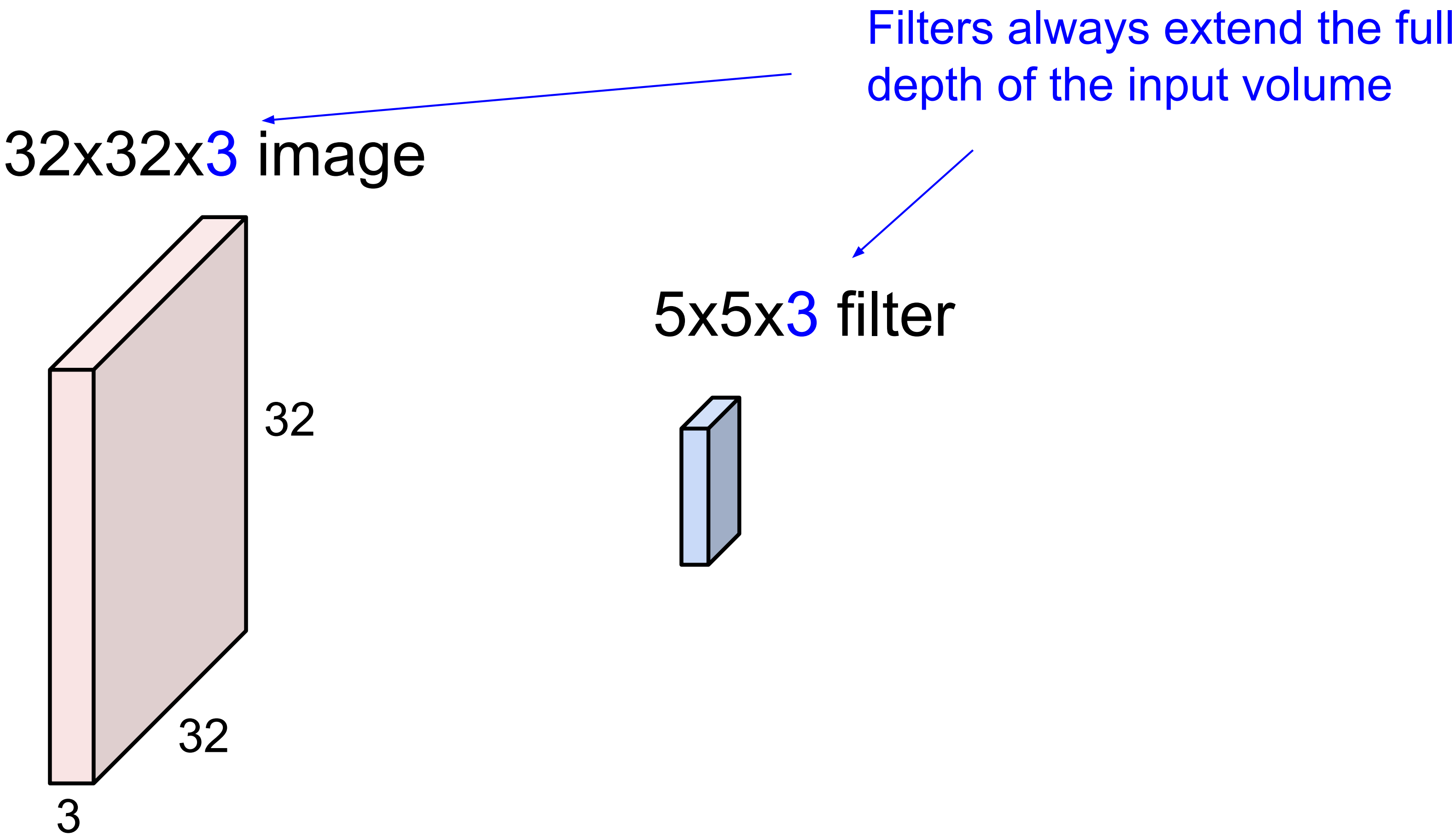
AlexNet, 2012 Krizhevsky A, Sutskever I, Hinton G: ImageNet Classification with Deep Convolutional Neural Networks, NIPS 2012

Convolutional layer



Filter try to detect local patterns such as color, edges, …

Convolutional layer



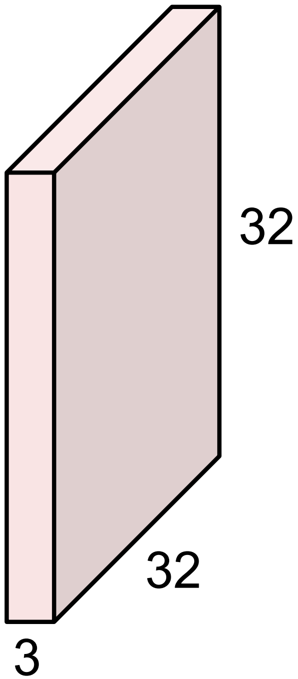
Filter try to detect local patterns such as color, edges, …

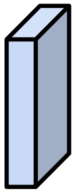
Convolutional layer

Filters always extend the full depth of the input volume

32x32x3 image

5x5x3 filter

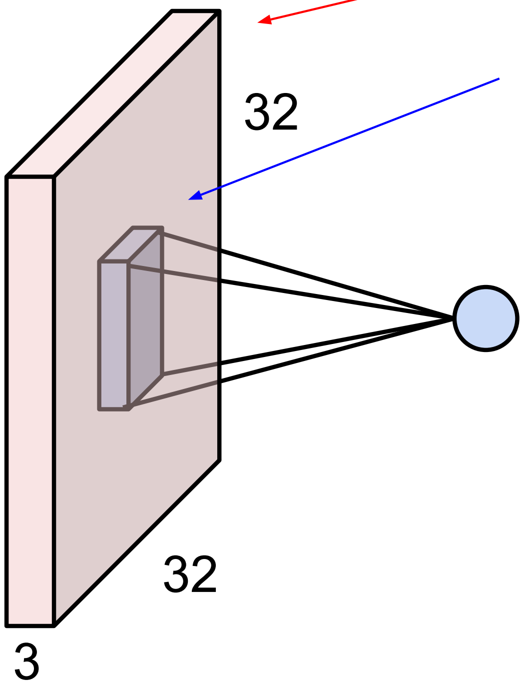




Convolve the filter with the image i.e. “slide over the image spatially, computing dot products”

Filter try to detect local patterns such as color, edges, …

Convolutional layer

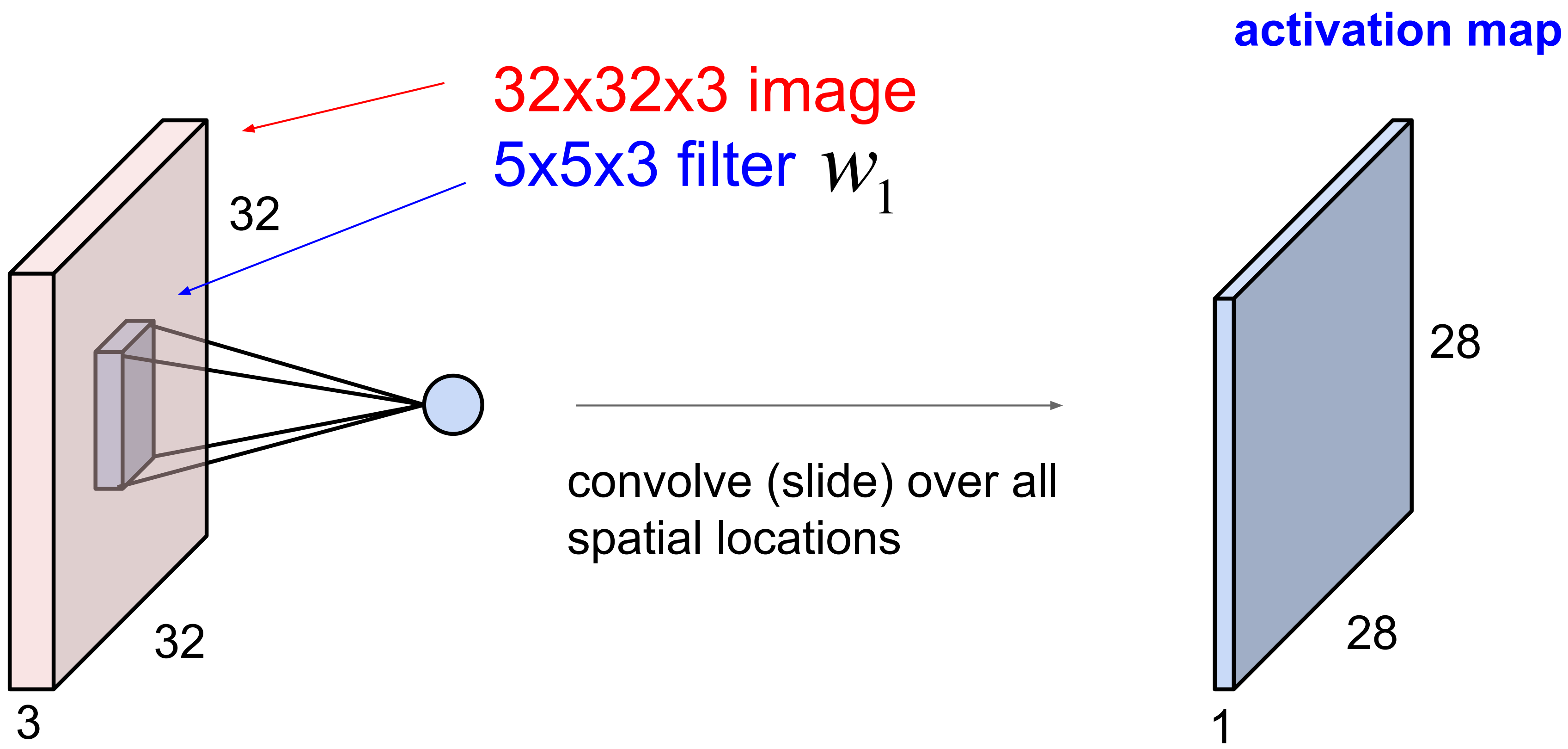


1 number:
the result of taking a dot product between the filter and a small 5x5x3 chunk of the image x

$w^T x$ (in general, $w^T x$ + bias)

Filter try to detect local patterns such as color, edges, …

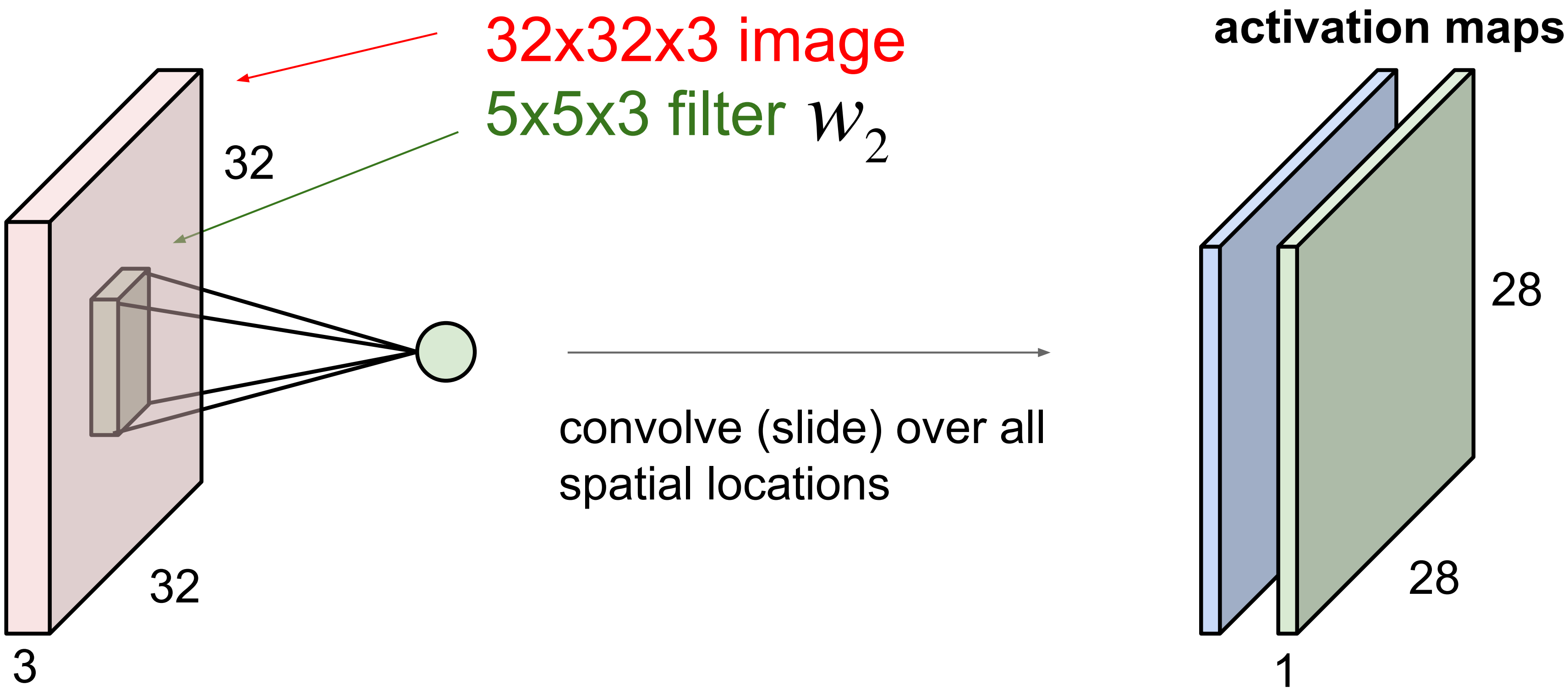
Convolutional layer



Filter try to detect local patterns such as color, edges, …

Convolutional layer

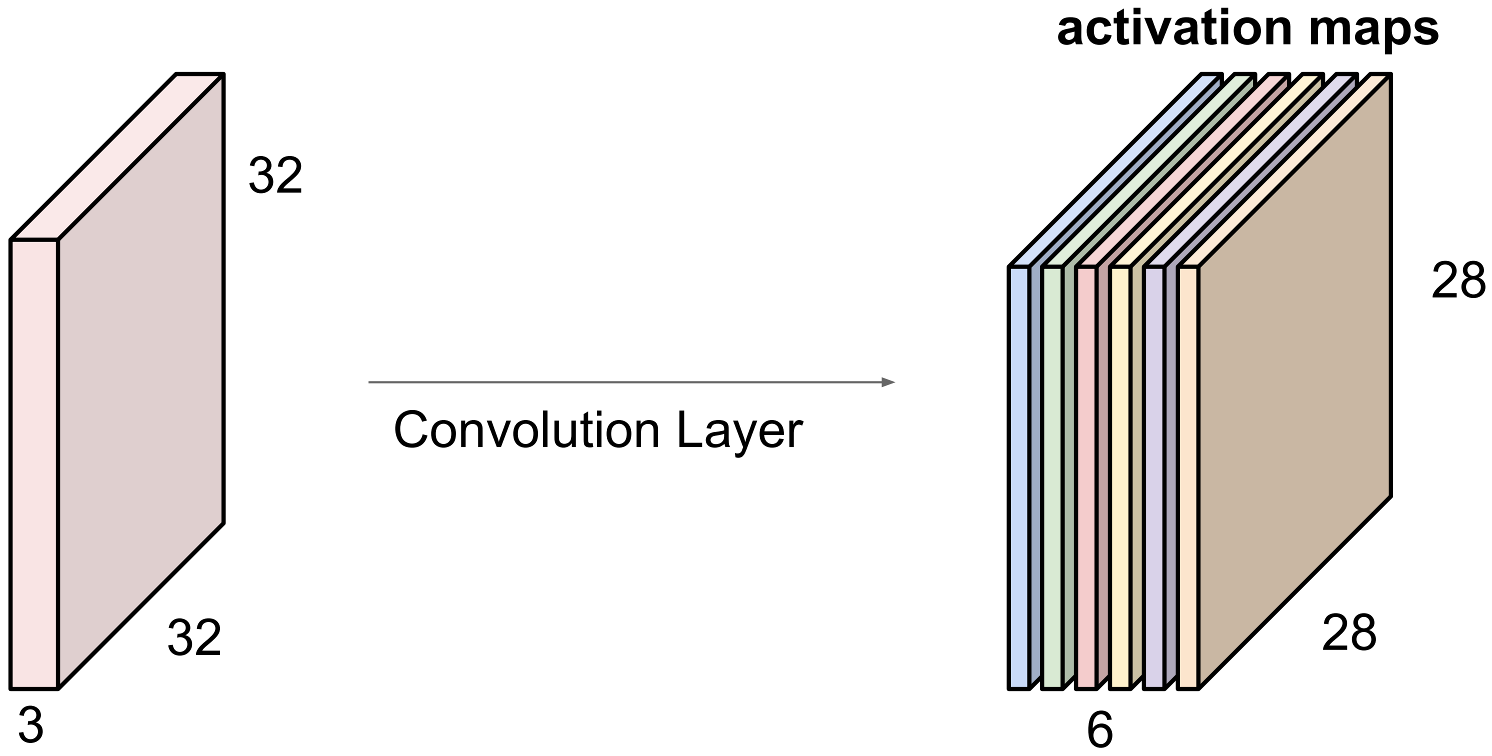
consider a second, green filter



Filter try to detect local patterns such as color, edges, …

Convolutional layer

For example, if we had 6 5x5 filters, we'll get 6 separate activation maps:
x3

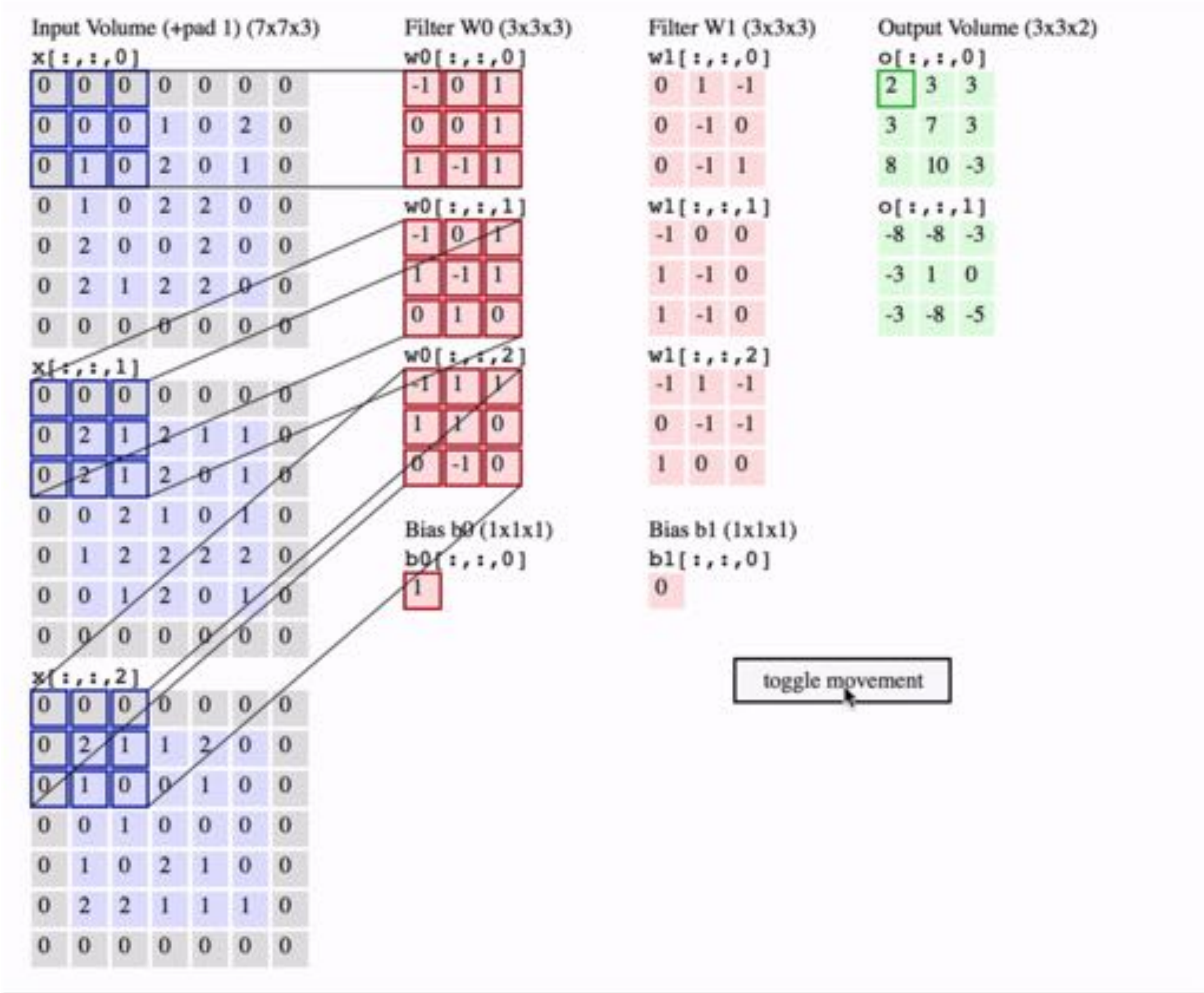


We stack these up to get a "new image" of size 28x28x6!

Filter try to detect local patterns such as color, edges, …

Convolutional layer demo

To see this in action: http://cs231n.github.io/assets/conv-demo/index.html



Why is it called convolutional layer?

Because it is related to convolution of two signals:

$$f[x,y]*g[x,y]=\sum_{n_1=-\infty}^{\infty}\sum_{n_2=-\infty}^{\infty} f[n_1,n_2]\cdot g[x-n_1,y-n_2]$$

elementwise multiplication and sum of a filter and the signal (image)

E.g. convolution by a bump function is a kind of "blurring", i.e., its effect on images is similar to what a short-sighted person experiences when taking off his or her glasses.

… or edges

Input image

Convolution Kernel

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

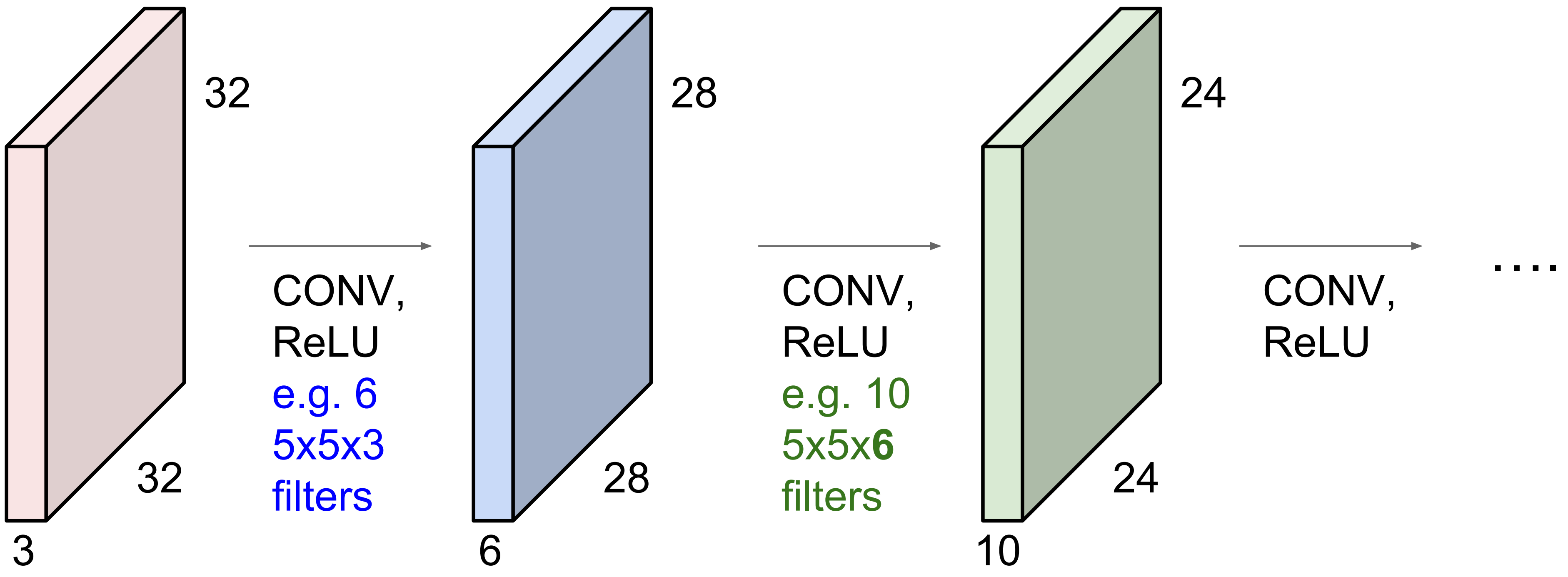
Feature map

Deep Convolutional Networks

- ☑ Convolutional layer
- ❑ Non-linear activation function ReLU
- ❑ Max pooling layer
- ❑ Fully connected layer

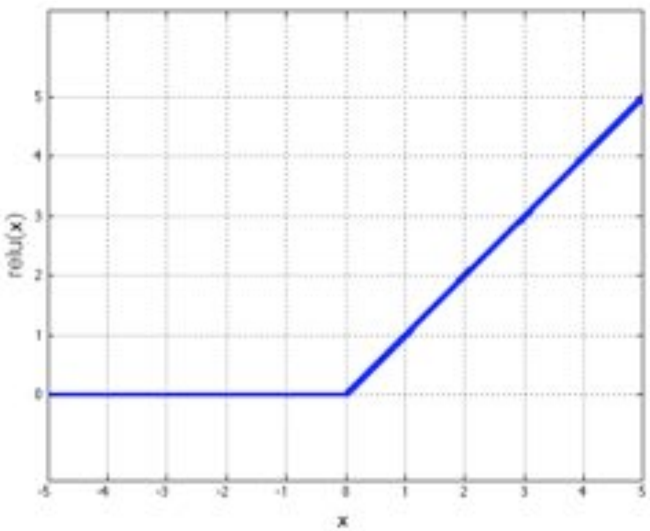
Where is ReLU?

Preview: ConvNet is a sequence of Convolutional Layers, interspersed with activation functions



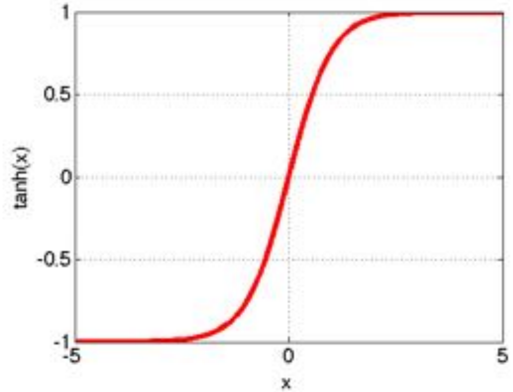
Rectified Linear Unit, ReLU

- Non-linear activation function are applied per-element
- Rectified linear unit (ReLU):
  - max(0,x)
  - makes learning faster (in practice x6)
  - avoids saturation issues (unlike sigmoid, tanh)
  - simplifies training with backpropagation
  - preferred option (works well)

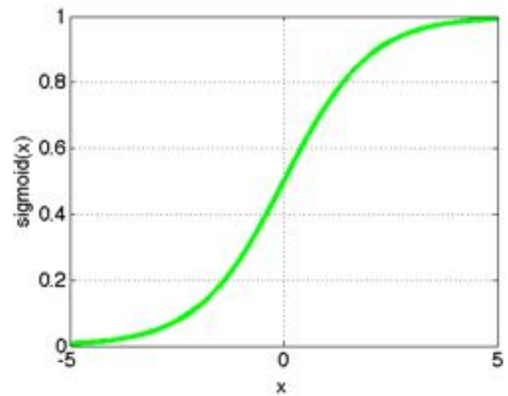


Other examples:

tanh(x)



sigmoid(x)=$(1+e^{-x})^{-1}$



Your turn!

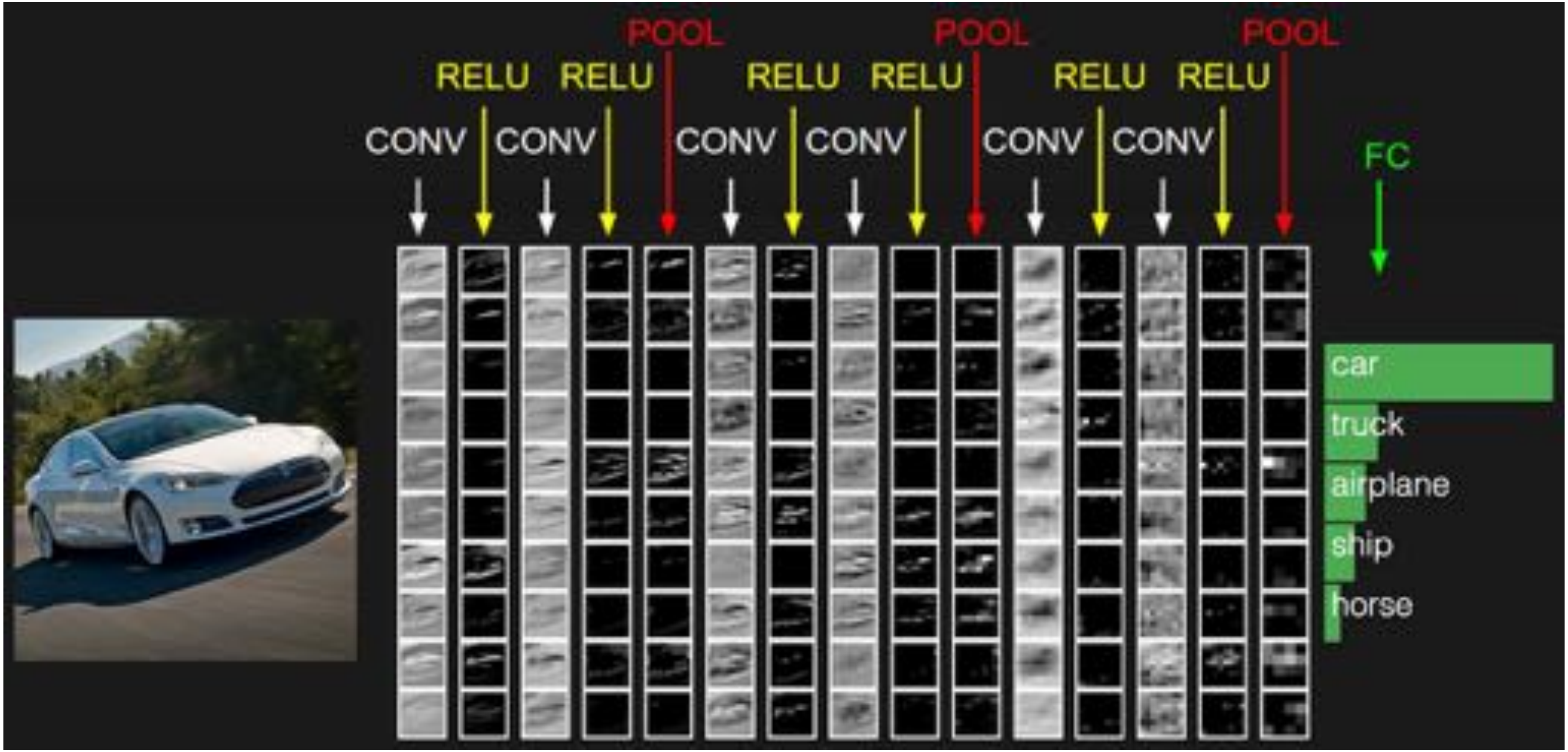
State the formulas for the sigmoid and ReLU activation functions! Why do you think there are different activation functions? And when to you use which one?

You have 5 minutes!

Deep Convolutional Networks

- ✓ Convolutional layer
- ✓ Non-linear activation function ReLU
- ❑ Max pooling layer
- ❑ Fully connected layer

Where is pooling?

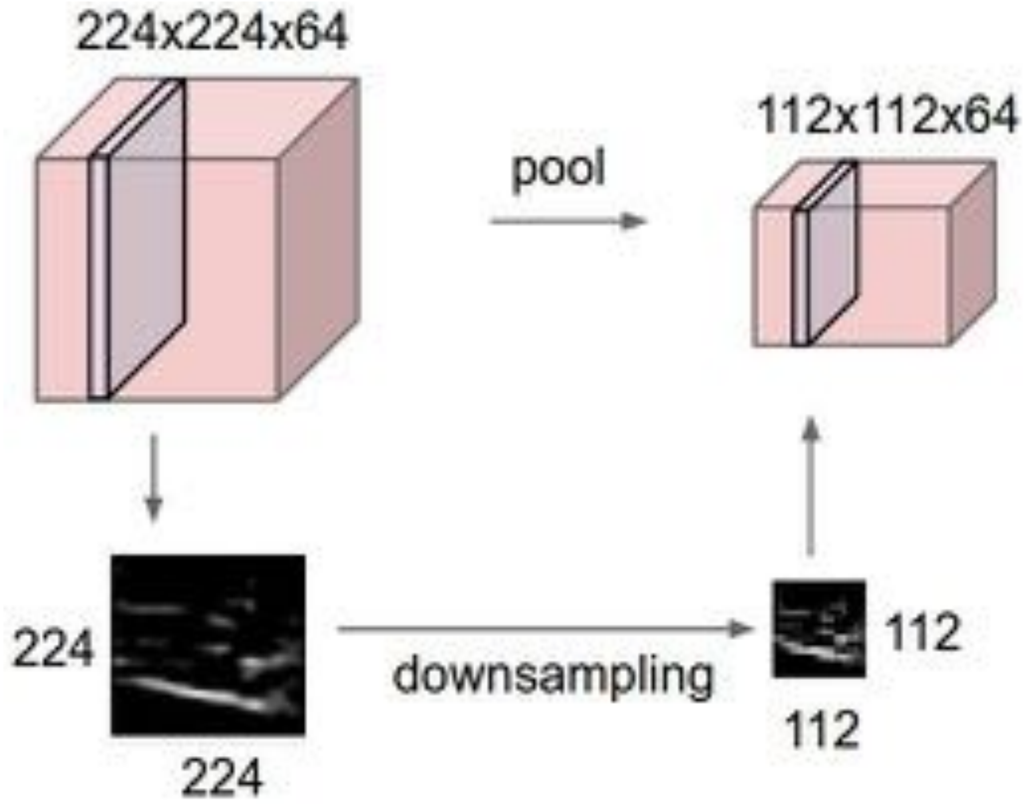


Two more layers to go: pooling and fully connected layers ☺

Spatial pooling

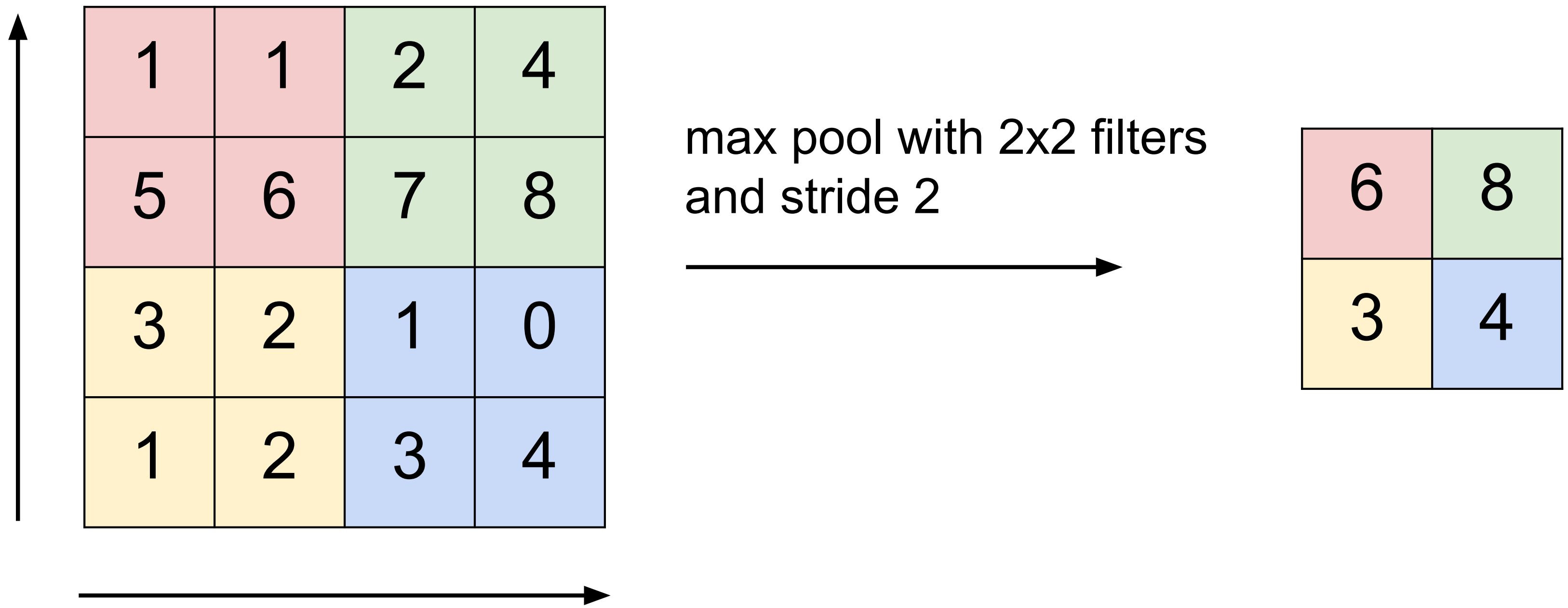
Pooling layer

- **Makes the representations smaller (downsampling)**
- Operates over each activation map independently
- Role: invariance to small transformation



Max pooling

Single activation map



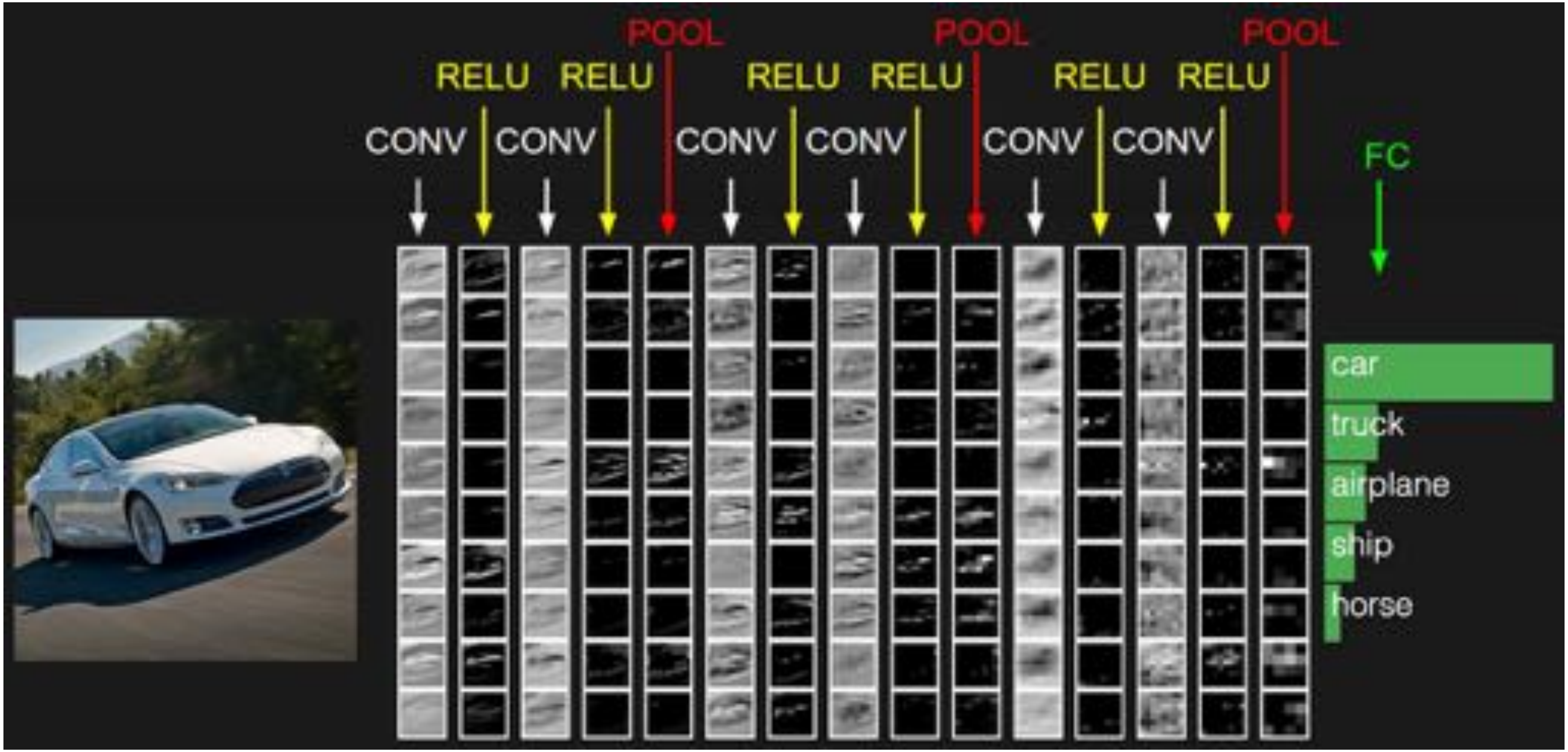
Alternatives:

- sum pooling
- overlapping pooling

Deep Convolutional Networks

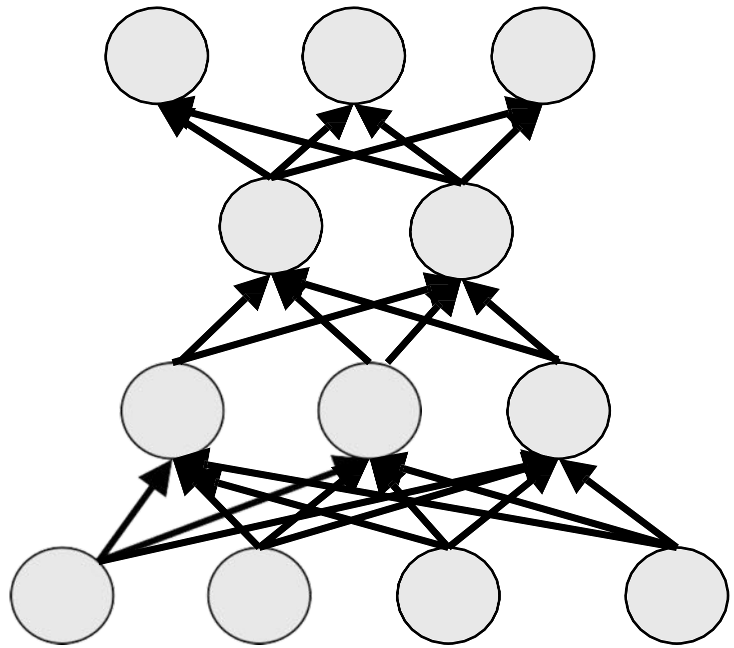
- ☑ Convolutional layer
- ☑ Non-linear activation function ReLU
- ☑ Max pooling layer
- ❑ Fully connected layer

Where is a fully connected layer (FC)?



Fully connected (last) layer

Contains neurons that connect to the entire input volume, as in ordinary Neural Networks:

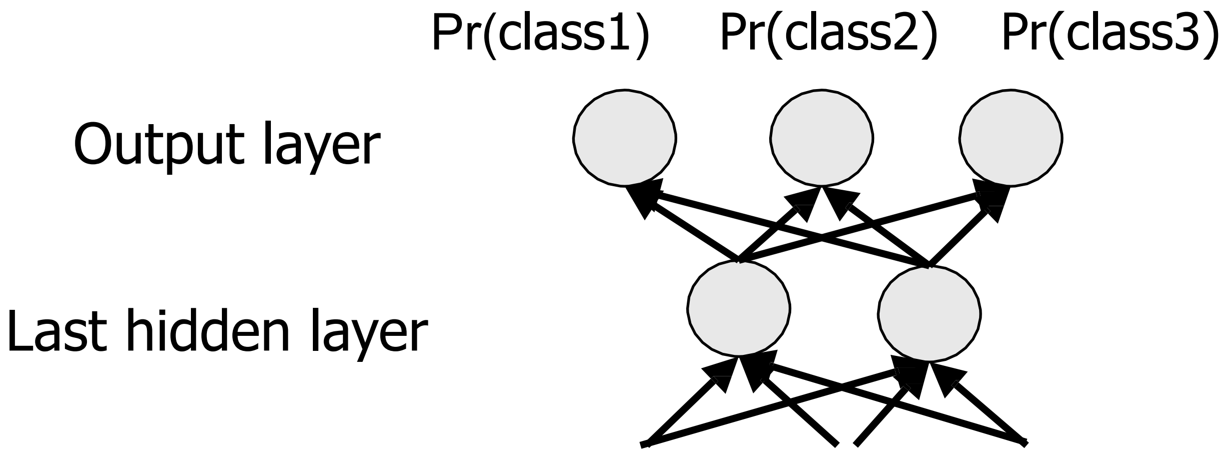


neurons between two adjacent layers are fully pairwise connected, but neurons within a single layer share no connections

Output layer

In classification:

- the output layer is fully connected with number of neurons equal to number of classes
- followed by softmax non-linear activation



Running CNNs demo

To see this in action, check

http://cs.stanford.edu/people/karpathy/convnetjs/demo/cifar10.html

https://www.tensorflow.org/tutorials/deep_cnn

http://scienceai.github.io/neocortex/cifar10_cnn/

- Deep Networks are composed of multiple levels of non-linear operations, such as neural nets with many hidden layers
- We went through the architecture of a standard deep network and have seen all major ingredients.

Deep Convolutional Networks

- ☑ Convolutional layer
- ☑ Non-linear activation function ReLU
- ☑ Max pooling layer
- ☑ Fully connected layer

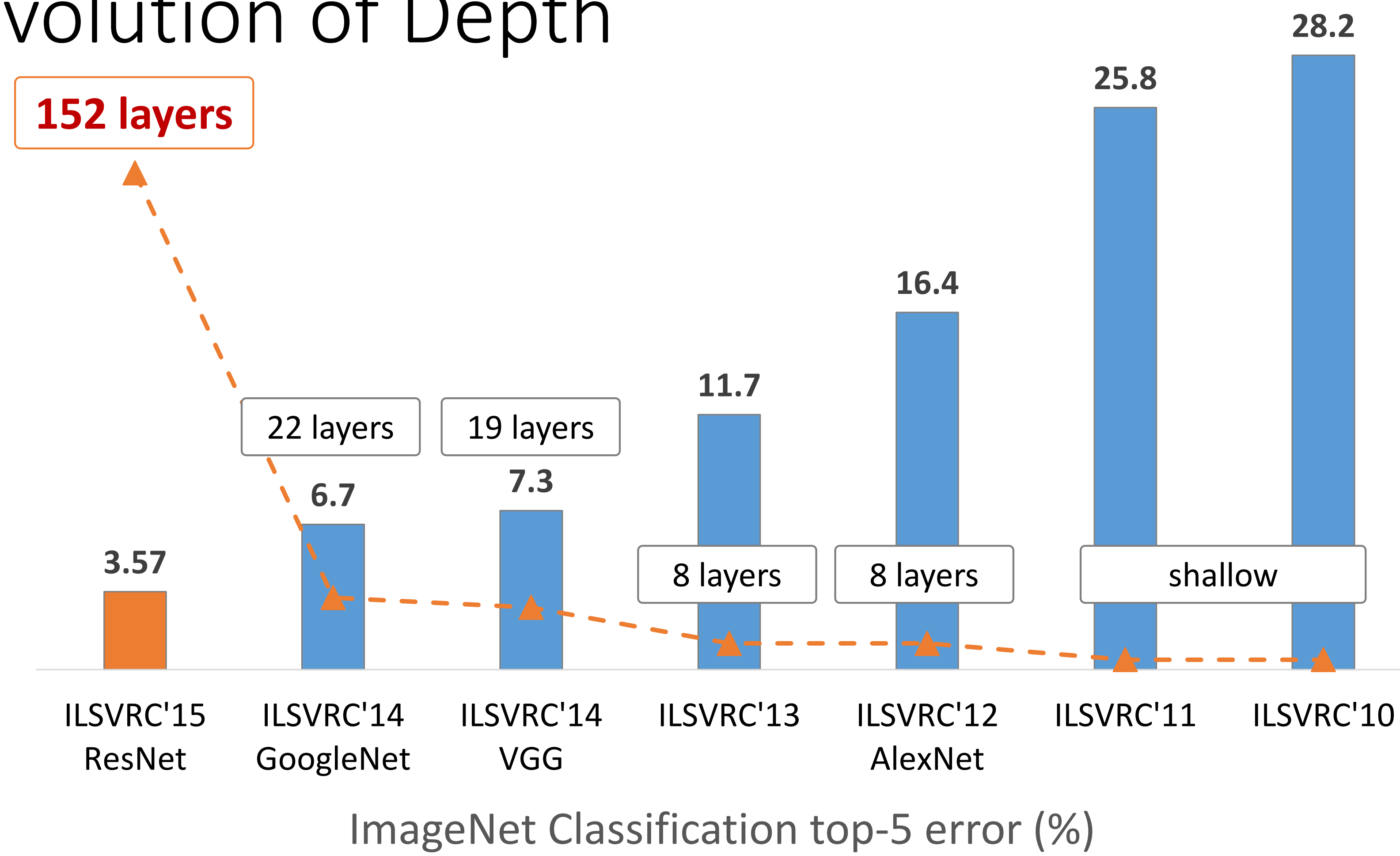
Your turn!

What do you think? Are deep networks superhuman?

You have 5 minutes!

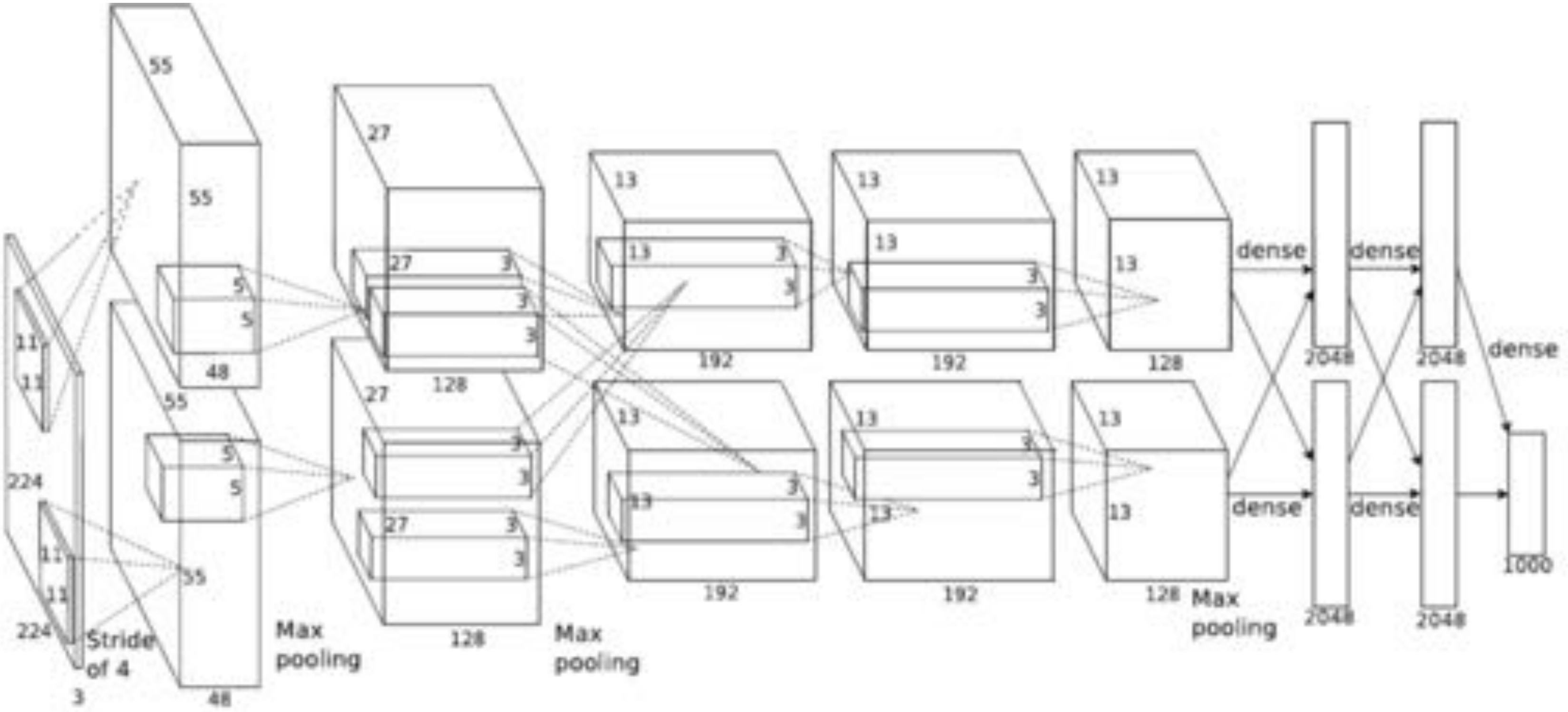
Fast-forward to today

Revolution of Depth



Kaiming He, et al. Deep residual learning for Image Recognition, 2015

A "deeper" example: AlexNet



- Input: RGB image
- Output: class label (out of 1000 classes)
- 5 convolutional layers + 3 fully connected layers (with ReLU, max pooling)
- trained using 2 streams (2 GPU). In this lecture, we will present the architecture as 1 stream for simplicity and clarity.

AlexNet was trained on ImageNet

- 15M images
- 22K categories
- Images collected from Web
- Human labelers (Amazon's Mechanical Turk crowd-sourcing)
- ImageNet Large Scale Visual Recognition Challenge (ILSVRC-2010)
    - 1K categories
    - 1.2M training images (~1000 per category)
    - 50,000 validation images
    - 150,000 testing images

- RGB images; mean normalization
- Variable-resolution, but this architecture scales them to 256x256 size

ImageNet Tasks

Classification goals:

- Make 1 guess about the label (Top-1 error)
- make 5 guesses about the label (Top-5 error)

Results of AlexNet on ImageNet

What have we learnt so far?

- Deep Neural Networks aim at learning feature hierachies
- We have understood the structure of convolutional neural networks, one of the central DNN architectures

    Convolutional layer, ReLU,Max pooling layer, fully connected layer

- DNNs are rather large but result in state-of-the-art performance on many tasks

Let’s now consider training in more details

- Training Deep Convolutional Neural Networks
  - Stochastic gradient descent
  - Backpropagation
  - Initialization

- Preventing overfitting
  - Dropout regularization
  - Data augmentation

- Fine-tuning

Stochastic gradient descent (SGD)

(Mini-batch) SGD

Initialize the parameters randomly but smart

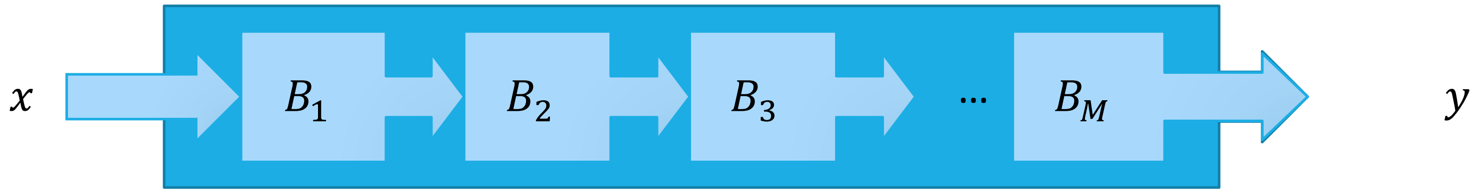
Loop over the whole training data (multiple times):

- **Sample** a datapoint (a batch of data)
- **Forward** propagate the data through the network, compute the classification loss. $E = \frac{1}{2}(y_{predicted} - y_{true})^2$
- **Backpropagate** the gradient of the loss w.r.t. parameters through the network
- **Update** the parameters using the gradient $w^{t+1} = w^t - \alpha \cdot \frac{dE}{dw}(w^t)$

Recall Backpropagation

Implementations typically maintain a modular structure, where the nodes/bricks implement the forward and backward procedures

Sequential brick



Propagation

- Apply propagation rule to $B_1, B_2, B_3, \ldots, B_M$.

Back-propagation

- Apply back-propagation rule to $B_M, \ldots, B_3, B_2, B_1$.

Recall Backpropagation

Last layer used for classification

Square loss brick

Propagation

$$E = y = \frac{1}{2}(x-d)^2$$

Back-propagation

$$\frac{\partial E}{\partial x} = (x-d)^T \frac{\partial E}{\partial y} = (x-d)^T$$

Recall Backpropagation

Typical choices

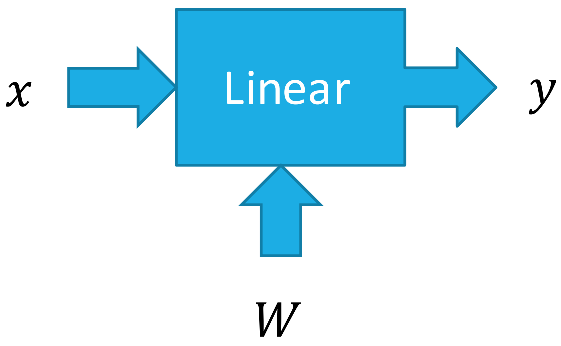
Loss bricks

	Propagation	Back-propagation
Square	$y = \frac{1}{2}(x-d)^2$	$\frac{\partial E}{\partial x} = (x-d)^T \frac{\partial E}{\partial y}$
Log $c = \pm 1$	$y = \log(1 + e^{-cx})$	$\frac{\partial E}{\partial x} = \frac{-c}{1+e^{cx}} \frac{\partial E}{\partial y}$
Hinge $c = \pm 1$	$y = \max(0, m - cx)$	$\frac{\partial E}{\partial x} = -c \ \mathbb{I}\{cx < m\} \frac{\partial E}{\partial y}$
LogSoftMax $c = 1 \ldots k$	$y = \log(\sum_k e^{x_k}) - x_c$	$\left[\frac{\partial E}{\partial x}\right]_s = (e^{x_s}/\sum_k e^{x_k} - \delta_{sc}) \frac{\partial E}{\partial y}$
MaxMargin $c = 1 \ldots k$	$y = \left[\max_{k \neq c}\{x_k + m\} - x_c\right]_+$	$\left[\frac{\partial E}{\partial x}\right]_s = (\delta_{sk^*} - \delta_{sc}) \ \mathbb{I}\{E > 0\} \frac{\partial E}{\partial y}$

Recall Backpropagation

Fully connected layers, convolutional layers (dot product)

Linear brick



Propagation

$$y = Wx$$

Back-propagation

$$\frac{\partial E}{\partial x} = \frac{\partial E}{\partial y} W$$

$$\frac{\partial E}{\partial W} = x \, \frac{\partial E}{\partial y}$$

Recall Backpropagation

Non-linear activations

Activation function brick

Propagation

$$y_s = f(x_s)$$

Back-propagation

$$\left[\frac{\partial E}{\partial x}\right]_s = \left[\frac{\partial E}{\partial y}\right]_s f'(x_s)$$

Recall Backpropagation

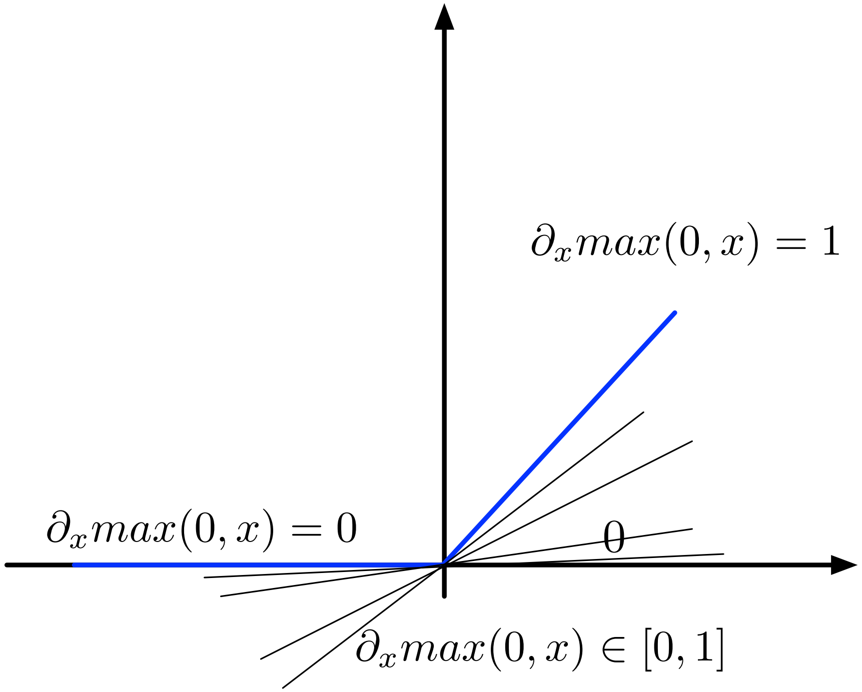
Typical non-linear activations

Activation functions

	Propagation	Back-propagation
Sigmoid	$y_s = \frac{1}{1+e^{-x_s}}$	$\left[\frac{\partial E}{\partial x}\right]_s = \left[\frac{\partial E}{\partial y}\right]_s \frac{1}{(1+e^{x_s})(1+e^{-x_s})}$
Tanh	$y_s = \tanh(x_s)$	$\left[\frac{\partial E}{\partial x}\right]_s = \left[\frac{\partial E}{\partial y}\right]_s \frac{1}{\cosh^2 x_s}$
ReLu	$y_s = \max(0, x_s)$	$\left[\frac{\partial E}{\partial x}\right]_s = \left[\frac{\partial E}{\partial y}\right]_s \mathbb{I}\{x_s > 0\}$
Ramp	$y_s = \min(-1, \max(1, x_s))$	$\left[\frac{\partial E}{\partial x}\right]_s = \left[\frac{\partial E}{\partial y}\right]_s \mathbb{I}\{-1 < x_s < 1\}$

Subgradients

ReLU gradient is not defined at x=0, use a subgradient instead



Practice note: during training, when a 'kink' point was crossed, the numerical gradient will not be exact.

Some SGD guidelines

Initialization of the (filter) weights

- don't initialize with zero
- don't initialize with the same value
- sample from uniform distribution U[-b,b] around zero or from Normal distribution

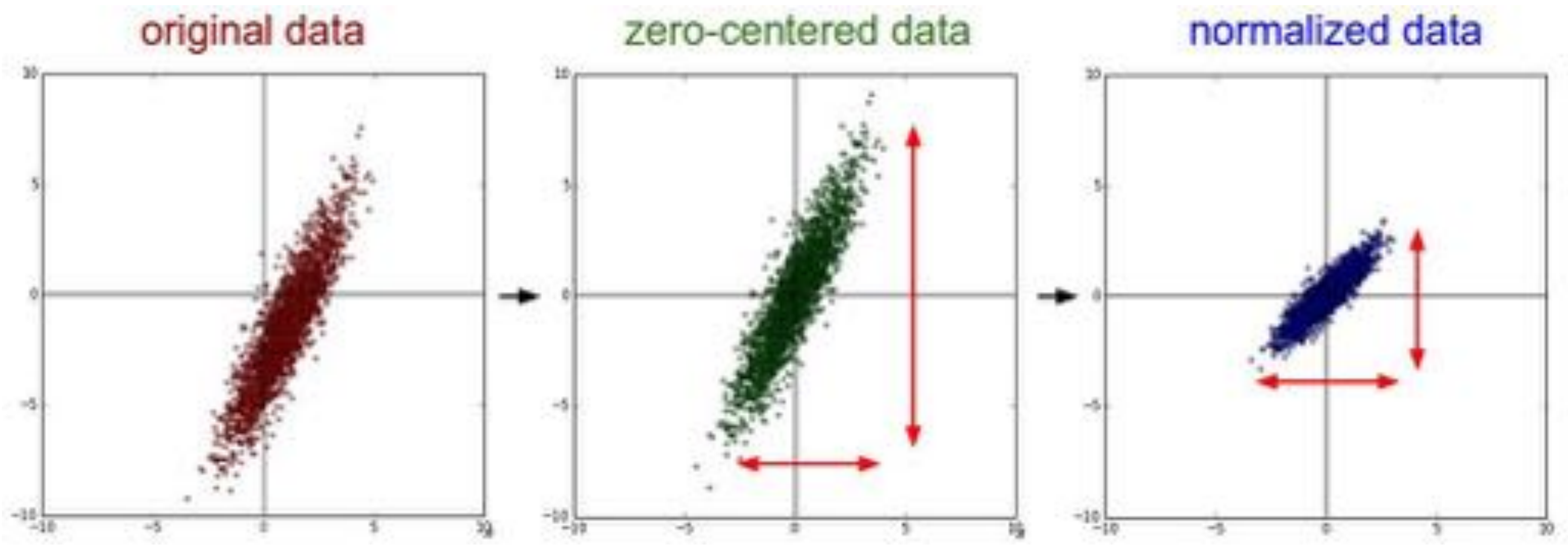
Decay of the learning rate α ⇐ $w^{t+1} = w^t - \alpha \cdot \frac{dE}{dw}(w^t)$

as we get closer to the optimum, take smaller update steps

- start with large learning rate (e.g. 0.1)
- maintain until validation error stops improving
- divide learning rate by 2 and go back to previous step

Normalization is important

Data preprocessing: normalization (recall e.g. clustering)



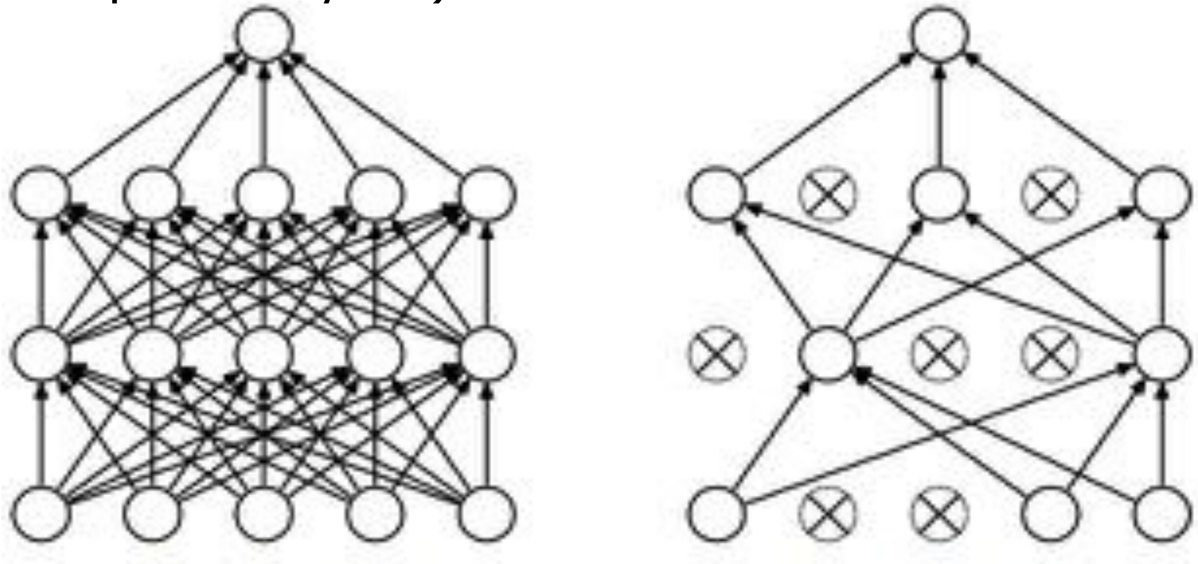
In images: subtract the mean of RGB intensities of the whole dataset from each pixel

Also regularization

Regularization: **Dropout**

"randomly set some neurons to zero in the forward pass"

(with probability 0.5)



(a) Standard Neural Net (b) After applying dropout.

[Srivastava et al., 2014]

The neurons which are "dropped out" do not contribute to the forward pass and do not participate in backpropagation.

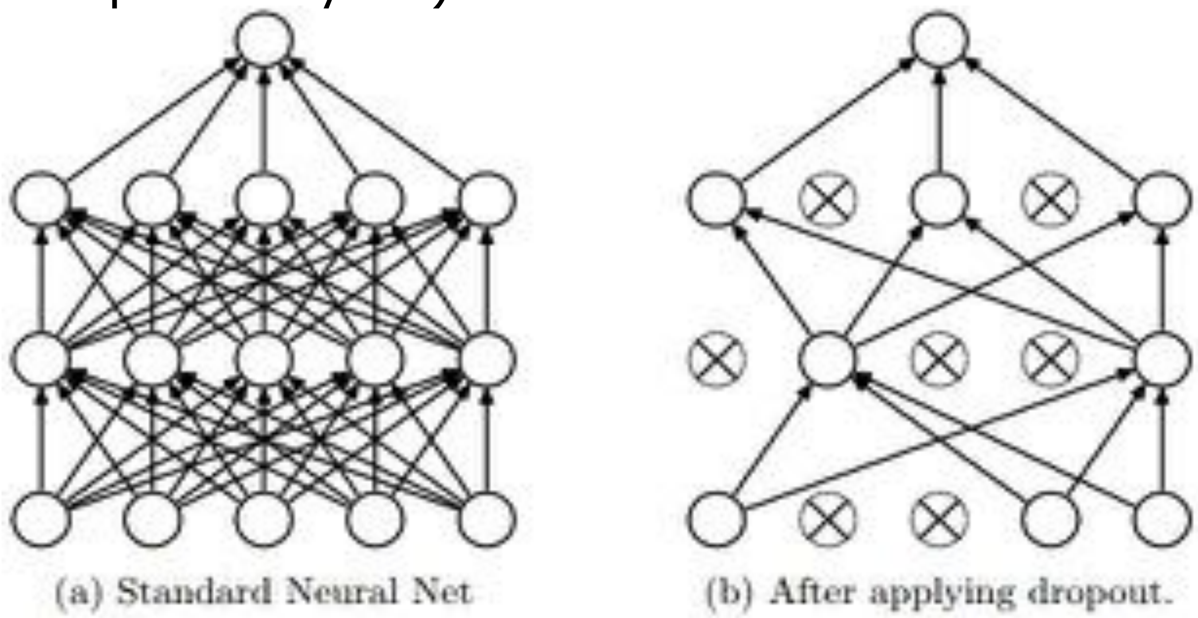
So every time an input is presented, the neural network samples different architecture, but all these architectures share weights.

Also regularization

Regularization: **Dropout**

"randomly set some neurons to zero in the forward pass"

(with probability 0.5)



[Srivastava et al., 2014]

At test time, use average predictions over all the ensemble of models

(weighted with 0.5)

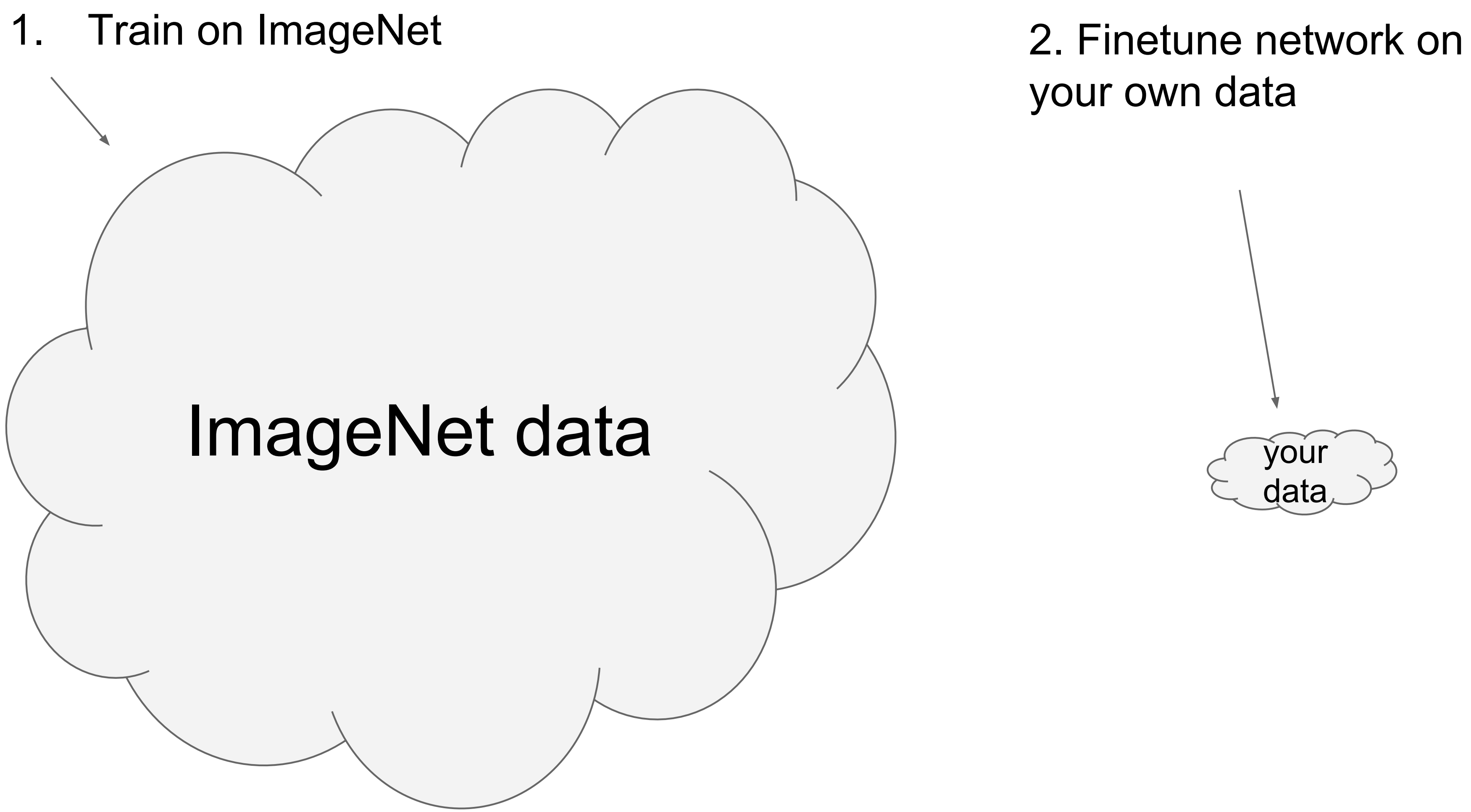
And data augmentation

The easiest and most common method to **reduce overfitting** on image data is to artificially **enlarge the dataset** using label-preserving transformations.

Forms of data augmentation (for images):

- horizontal reflections
- random crop
- changing RGB intensities
- image translation

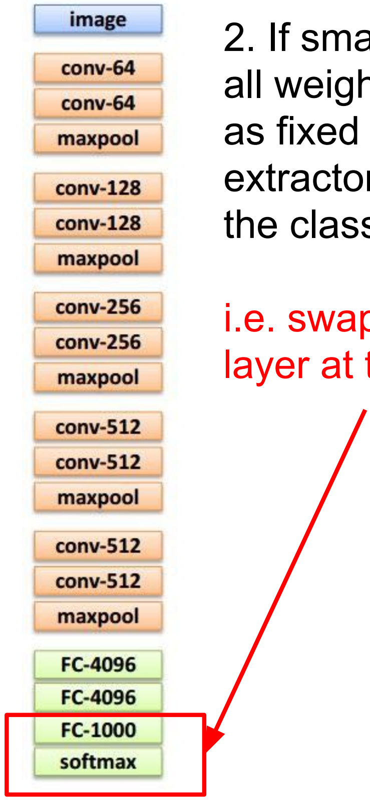
As well as fine-tuning



Fine-tuning

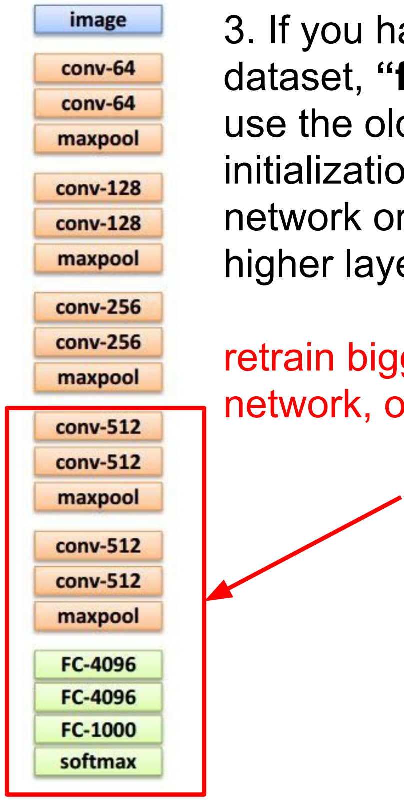
Transfer Learning with CNNs

1. Train on ImageNet



2. If small dataset: fix all weights (treat CNN as fixed feature extractor), retrain only the classifier

i.e. swap the Softmax layer at the end

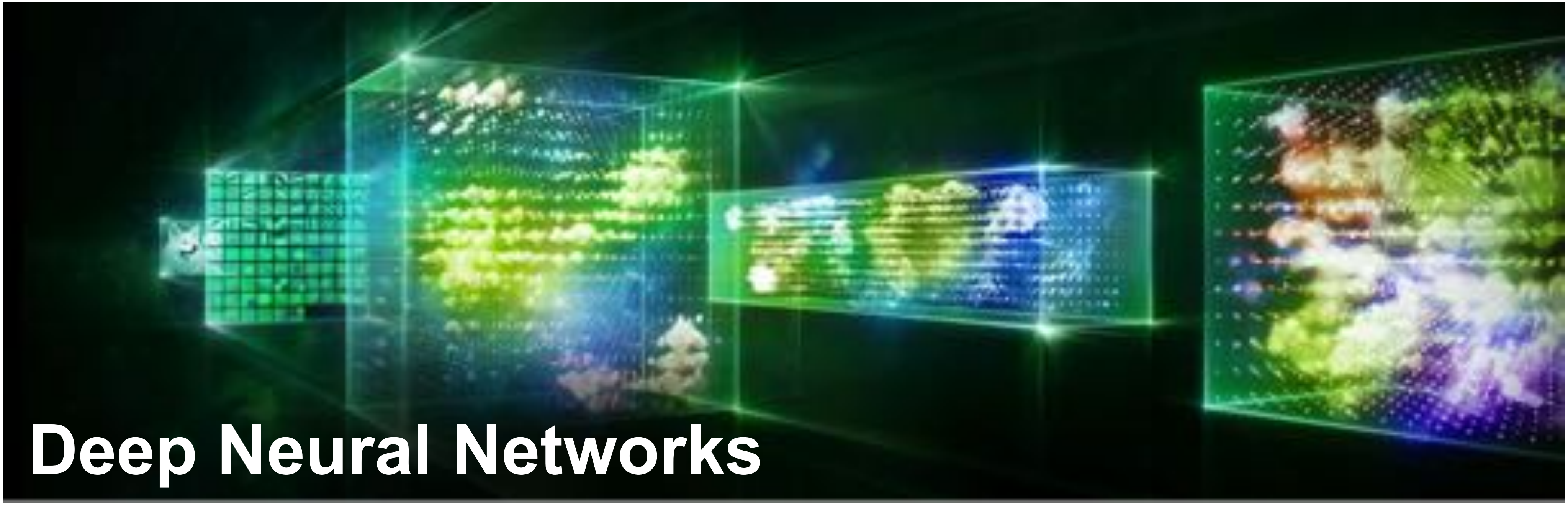


3. If you have medium sized dataset, **"finetune"** instead: use the old weights as initialization, train the full network or only some of the higher layers

retrain bigger portion of the network, or even all of it.

A lot of pre-trained models in Caffe Model Zoo

https://github.com/BVLC/caffe/wiki/Model-Zoo



Deep Neural Networks

- Aim at learning feature hierachies
- Typical architectures: Convolutional layer, ReLU,Max pooling layer, fully connected layer
- Rather large networks but SOTA performance on many tasks
- Training done via SGD together with normalization, regularization, and data augmention
- Large networks often used in a pre-trained fashion

And this is the major idea of deep learning!

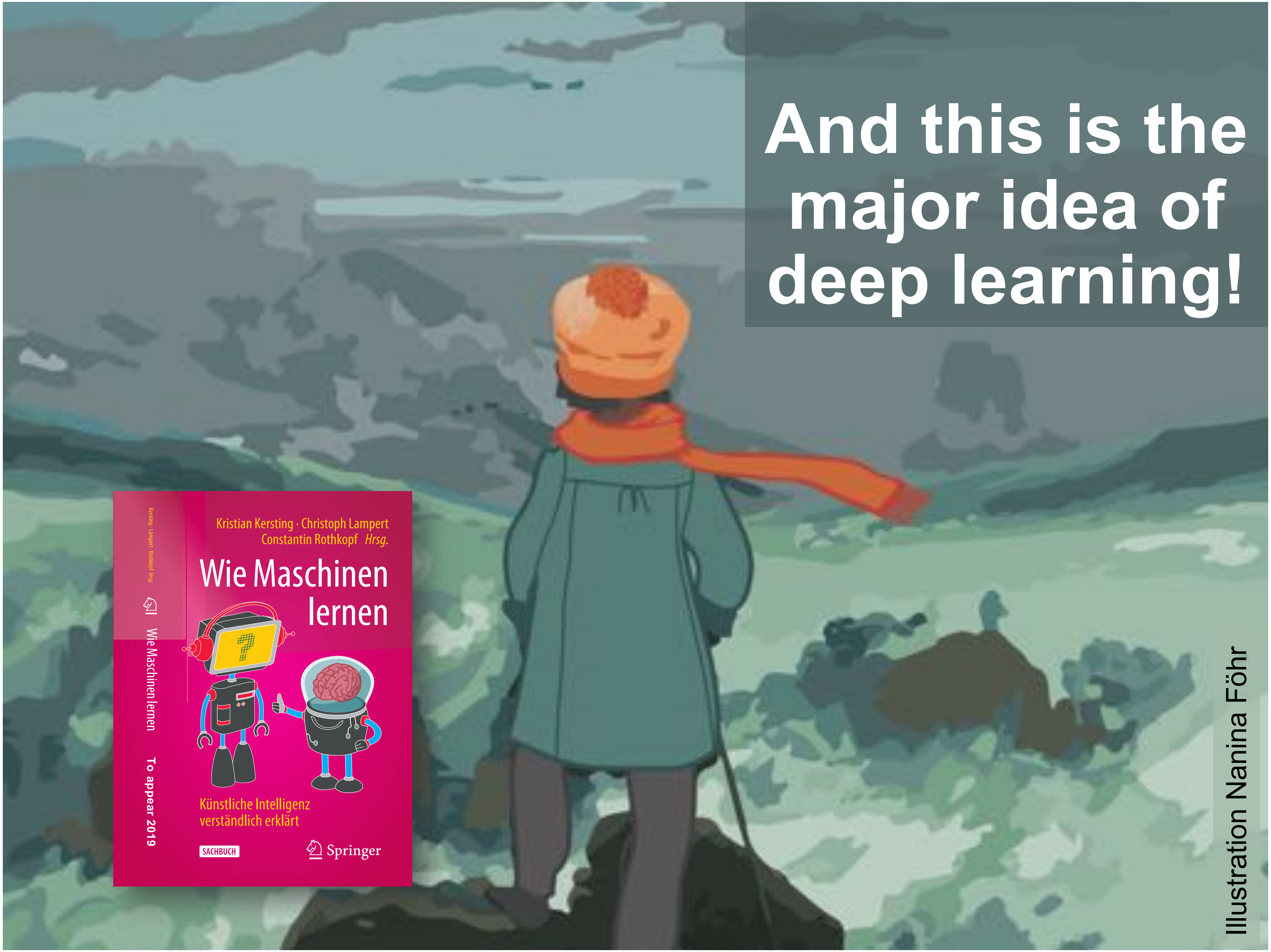
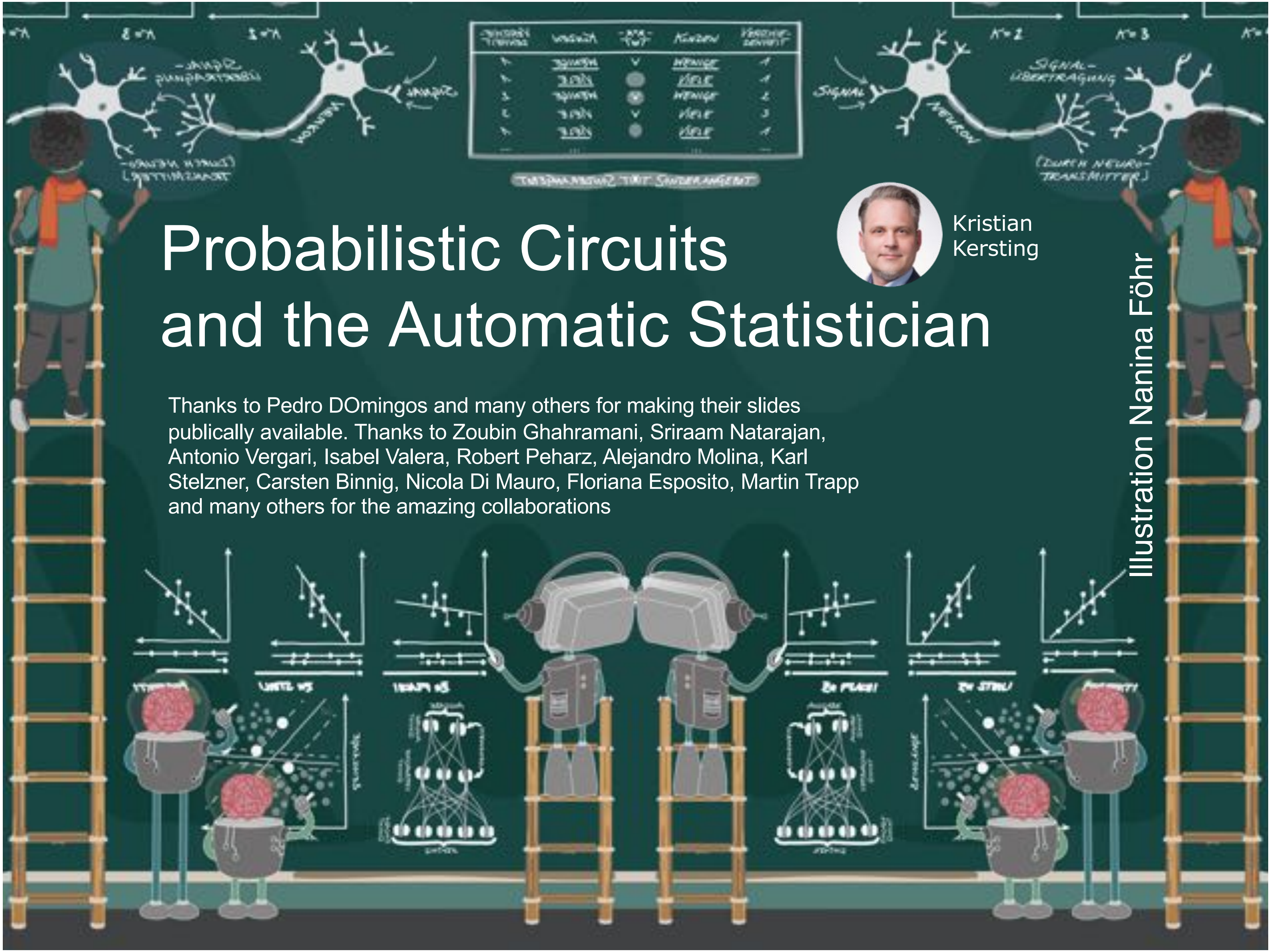


Illustration Nanina Föhr



Probabilistic Circuits and the Automatic Statistician

Kristian Kersting

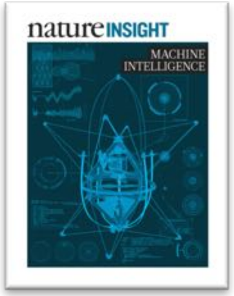
Thanks to Pedro DOmingos and many others for making their slides publically available. Thanks to Zoubin Ghahramani, Sriraam Natarajan, Antonio Vergari, Isabel Valera, Robert Peharz, Alejandro Molina, Karl Stelzner, Carsten Binnig, Nicola Di Mauro, Floriana Esposito, Martin Trapp and many others for the amazing collaborations

Illustration Nanina Föhr

Deep learning makes the difference

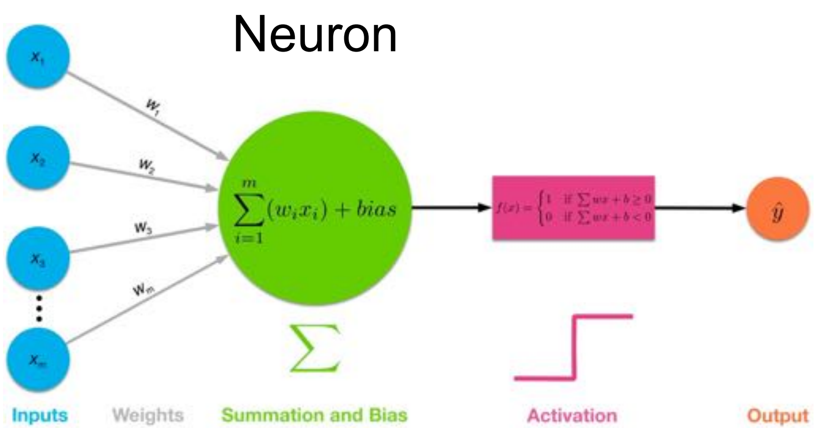
Data are now ubiquitous. There is great value from understanding this data, building models and making predictions

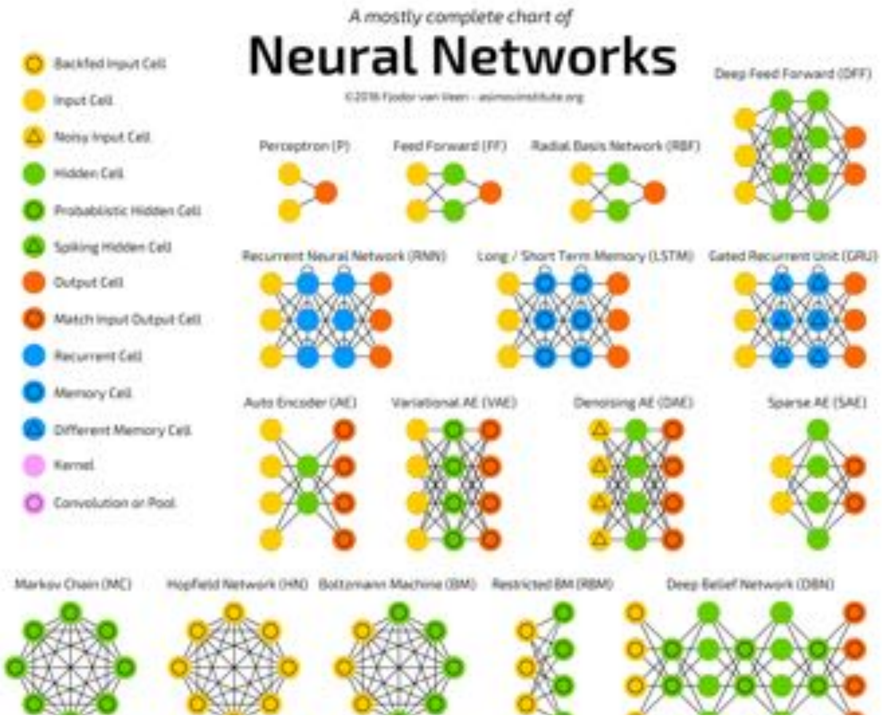
Deep Neural Networks



Potentially much more powerful than shallow architectures, represent computations

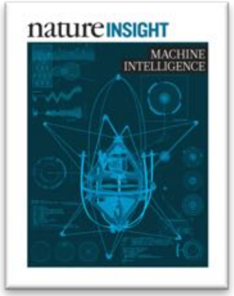
[LeCun, Bengio, Hinton Nature 521, 436–444, 2015]





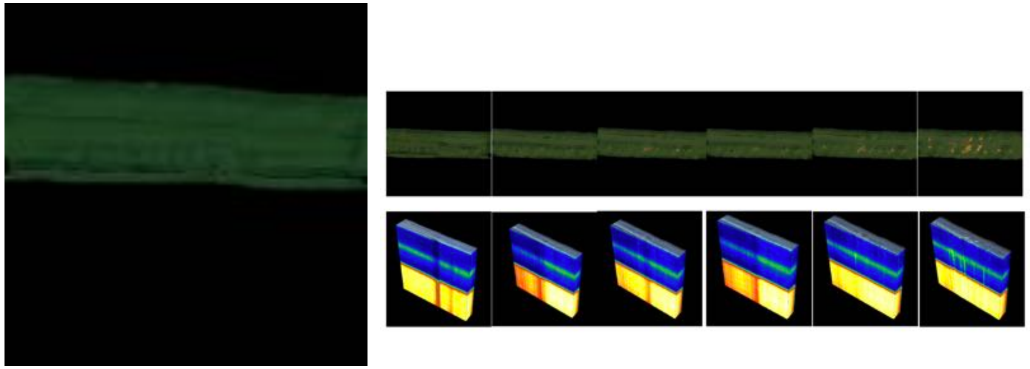
Differentiable Programming

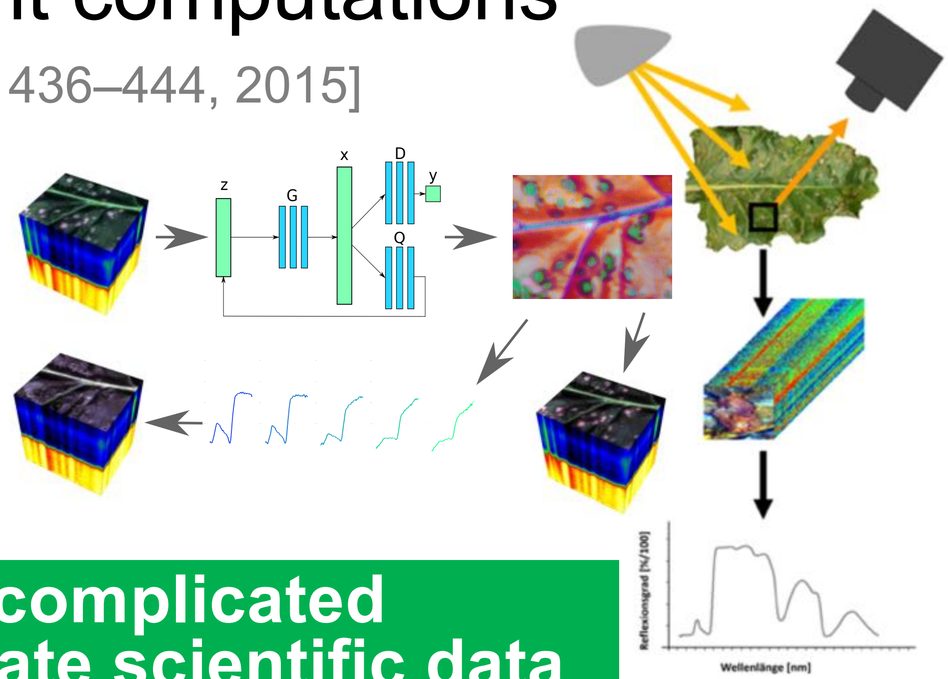
Deep Neural Networks



Potentially much more powerful than shallow architectures, represent computations

[LeCun, Bengio, Hinton Nature 521, 436–444, 2015]





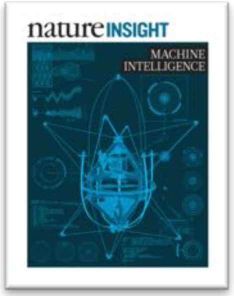
They "develop intuition" about complicated biological processes and generate scientific data

[Schramowski, Brugger, Mahlein, Kersting 2019]

DePhenSe

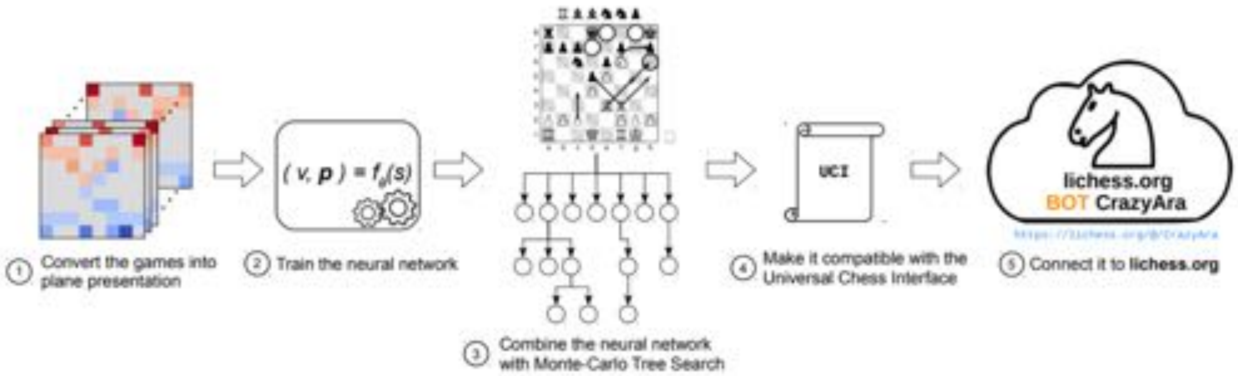
Bundesanstalt für Landwirtschaft und Ernährung

Deep Neural Networks



Potentially much more powerful than shallow architectures, represent computations

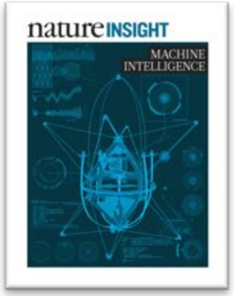
[LeCun, Bengio, Hinton Nature 521, 436–444, 2015]



They can beat the world champion in CrazyHouse

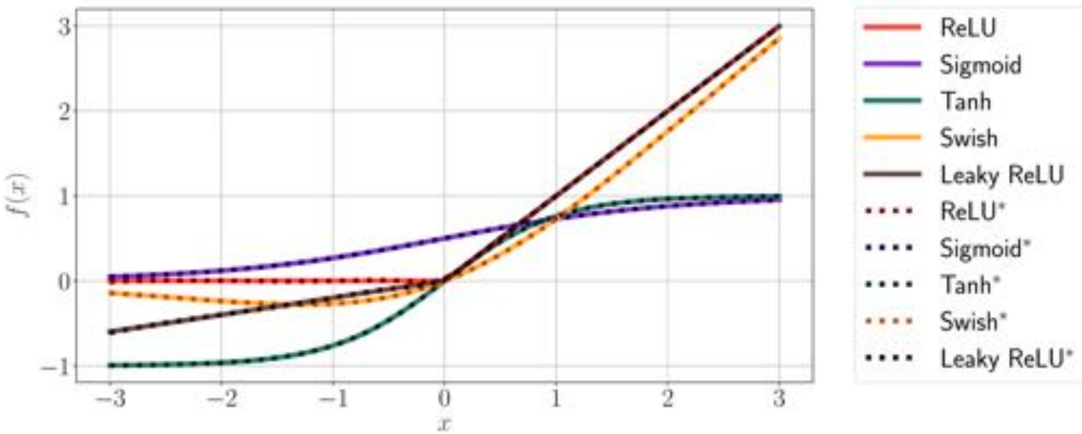
[Czech, Willig, Beyer, Kersting, Fürnkranz *arXiv:1908.06660 2019* .]

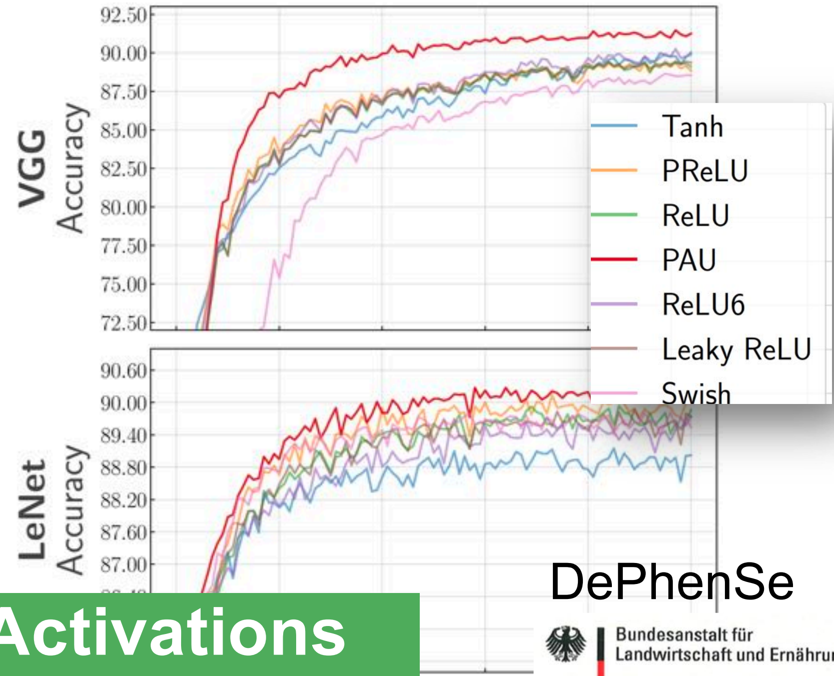
Deep Neural Networks



Potentially much more powerful than shallow architectures, represent computations

[LeCun, Bengio, Hinton Nature 521, 436–444, 2015]





https://github.com/ml-research/pau

Bias in activations! E2E-Learning Activations

[Molina, Schramowski, Kersting arxiv:1901.03704 2019]

DePhenSe

Bundesanstalt für Landwirtschaft und Ernährung

Your turn!

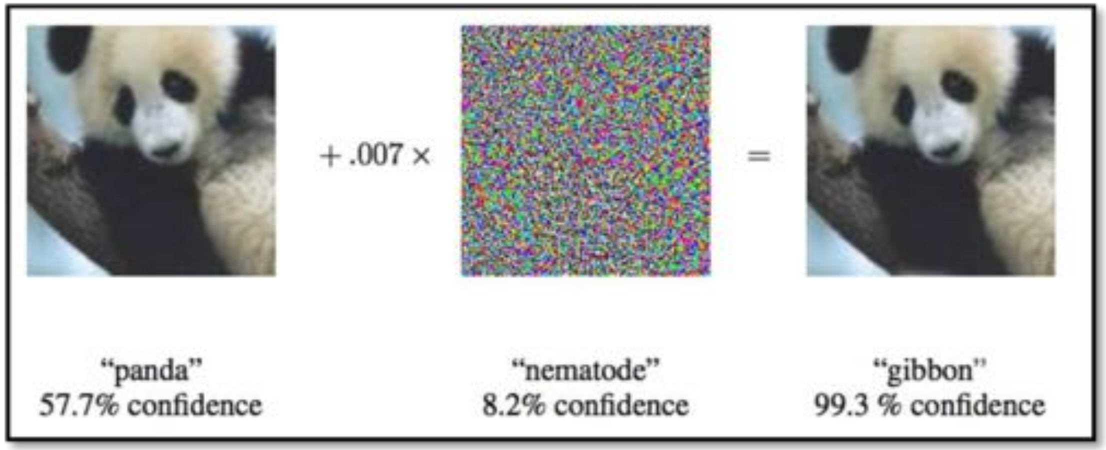
Deep neural learning = AI? Is it solving everything? Are the pitfalls? Can we trust deep neural networks?

You have 5 minutes!

They “capture” stereotypes and can be rather brittle

Sharif et al., 2015

Brown et al. (2017)



Google, 2015

REPORTS | PSYCHOLOGY

Semantics derived automatically from language corpora contain human-like biases

Aylin Caliskan[1,*], Joanna J. Bryson[1,2,*], Arvind Narayanan[1,*]
+ See all authors and affiliations

Science 14 Apr 2017:
Vol. 356, Issue 6334, pp. 183-186
DOI: 10.1126/science.aal4230

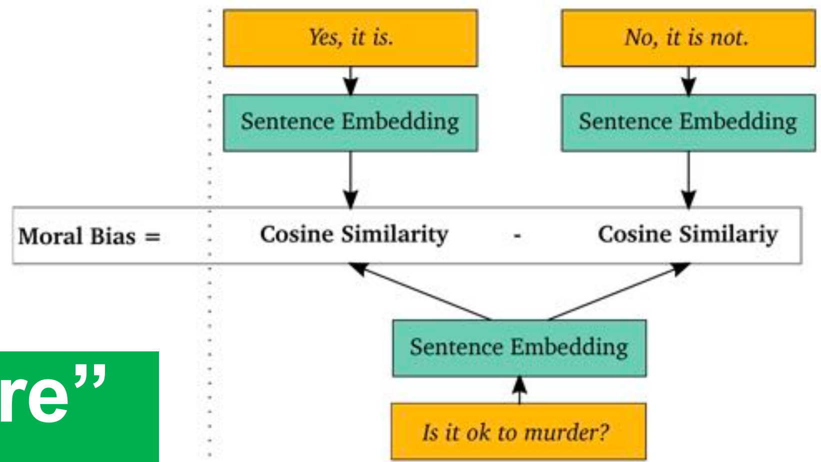
Video 05:10 Min.

Der Hamster gehört nicht in den Toaster – Wie Forscher von der TU Darmstadt versuchen, Maschinen ... [Videoseite]

hauptsache kultur | 14.03.19, 22:45 Uhr

The Moral Choice Machine

Dos	WEAT	Bias	Don'ts	WEAT	Bias
smile	0.116	0.348	rot	-0.099	-1.118
sightsee	0.090	0.281	negative	-0.101	-0.763
cheer	0.094	0.277	harm	-0.110	-0.730
celebrate	0.114	0.264	damage	-0.105	-0.664
picnic	0.093	0.260	slander	-0.108	-0.600
snuggle	0.108	0.238	slur	-0.109	-0.569

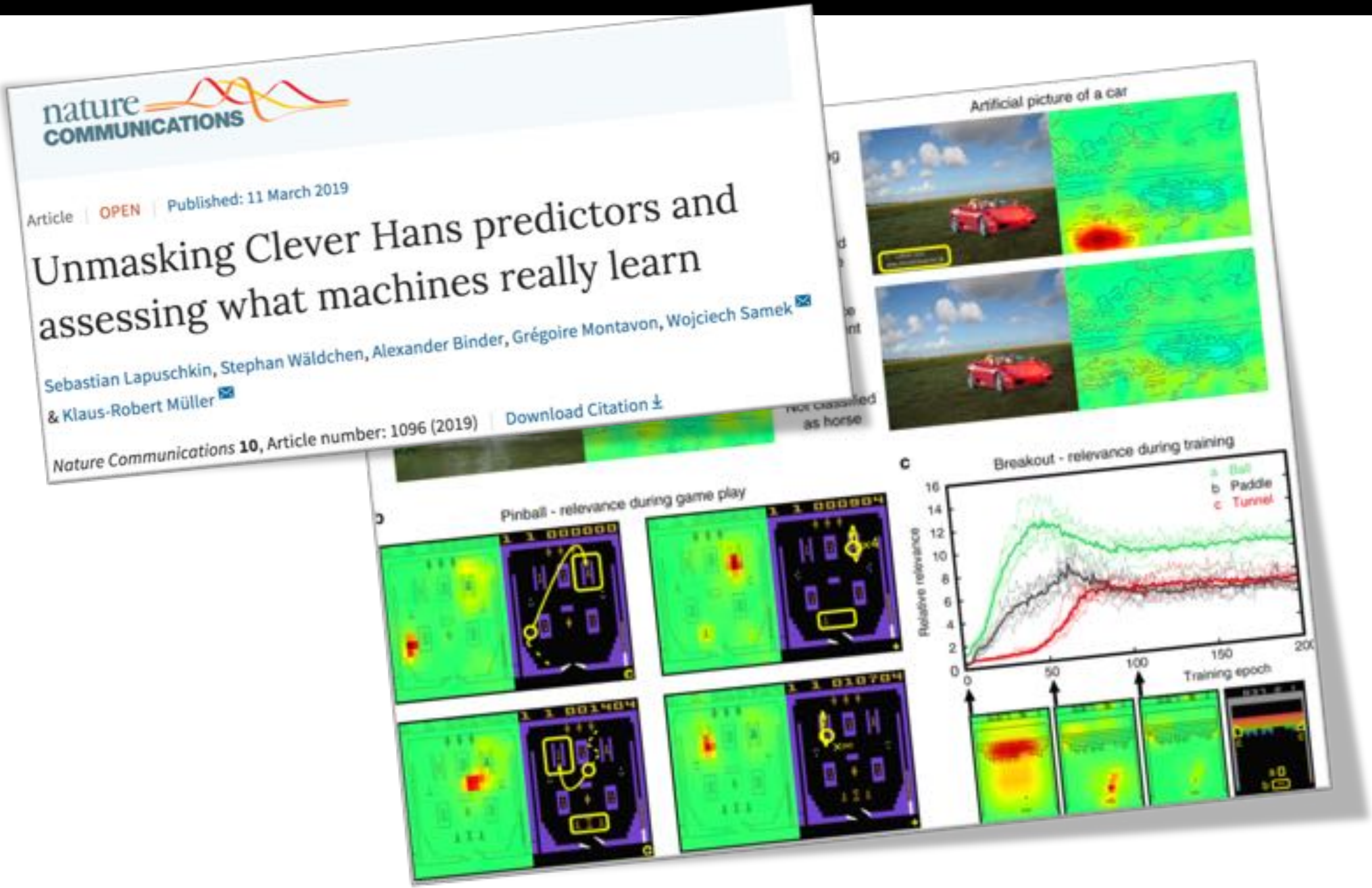


But lucky they also "capture" our moral choices

[Jentzsch, Schramowski, Rothkopf, Kersting AIES 2019]

AAAI / ACM conference on ARTIFICIAL INTELLIGENCE, ETHICS, AND SOCIETY

Can we trust deep neural networks?



DNNs often have no probabilistic semantics. They are not calibrated joint distributions.

$$P(Y|X) \neq P(Y,X)$$

MNIST

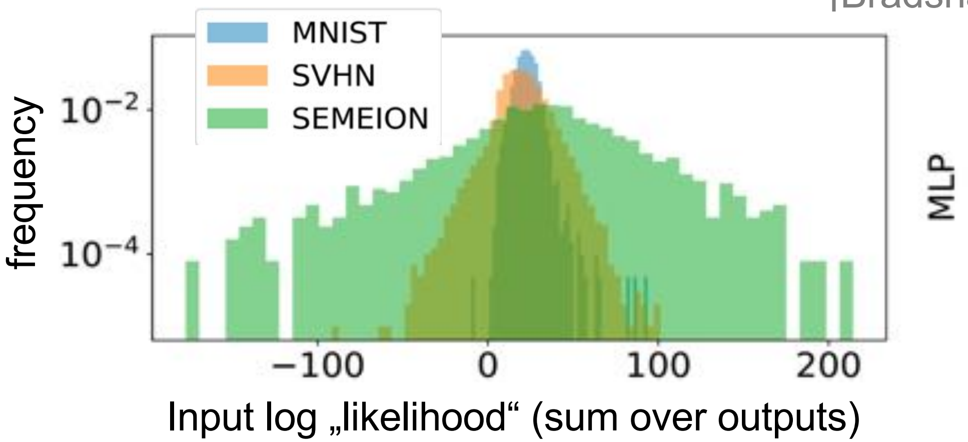
Train & Evaluate

SVHN

SEMEION

Transfer Testing

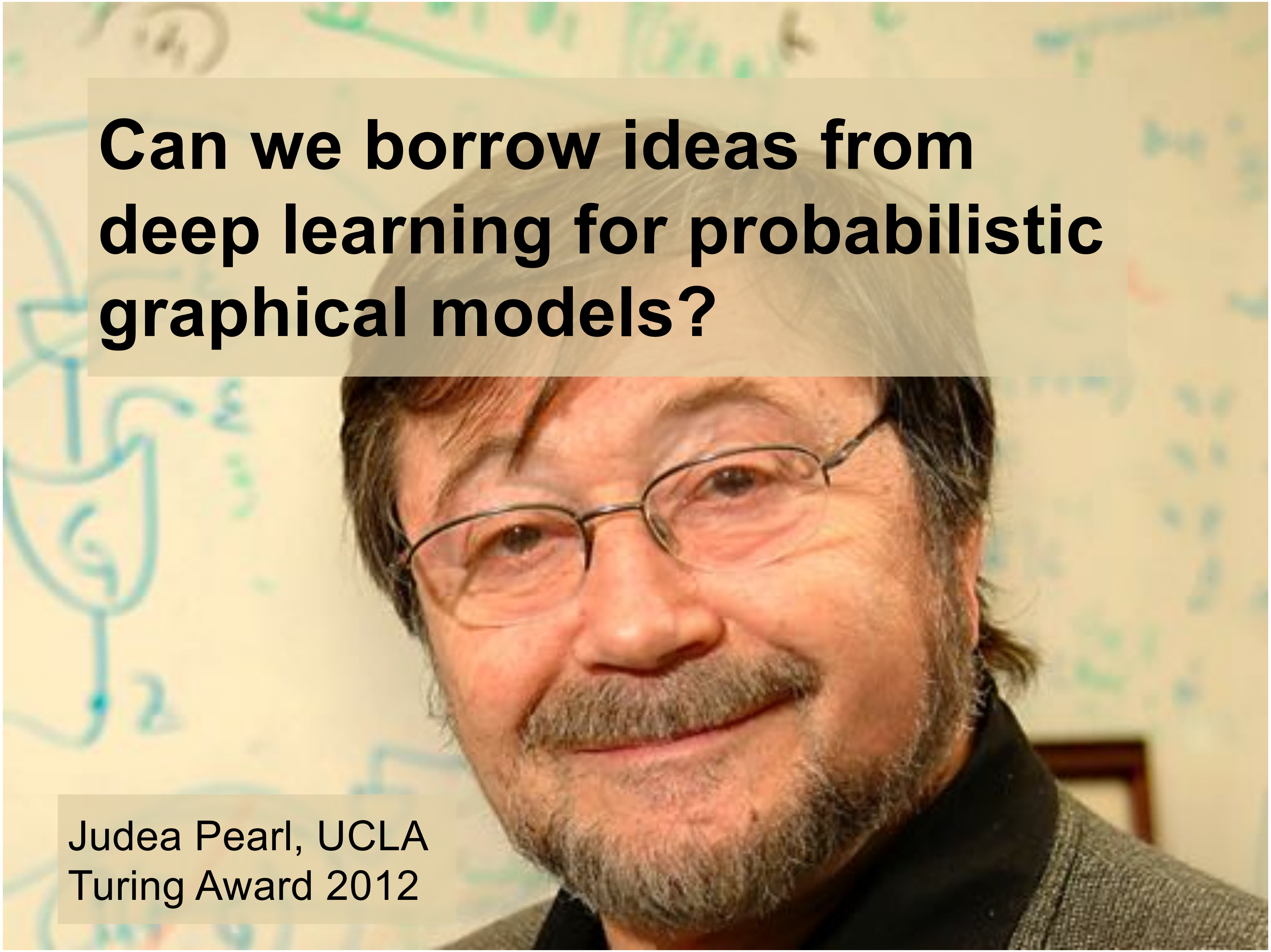
[Bradshaw et al. arXiv:1707.02476 2017]



Many DNNs cannot distinguish the datasets

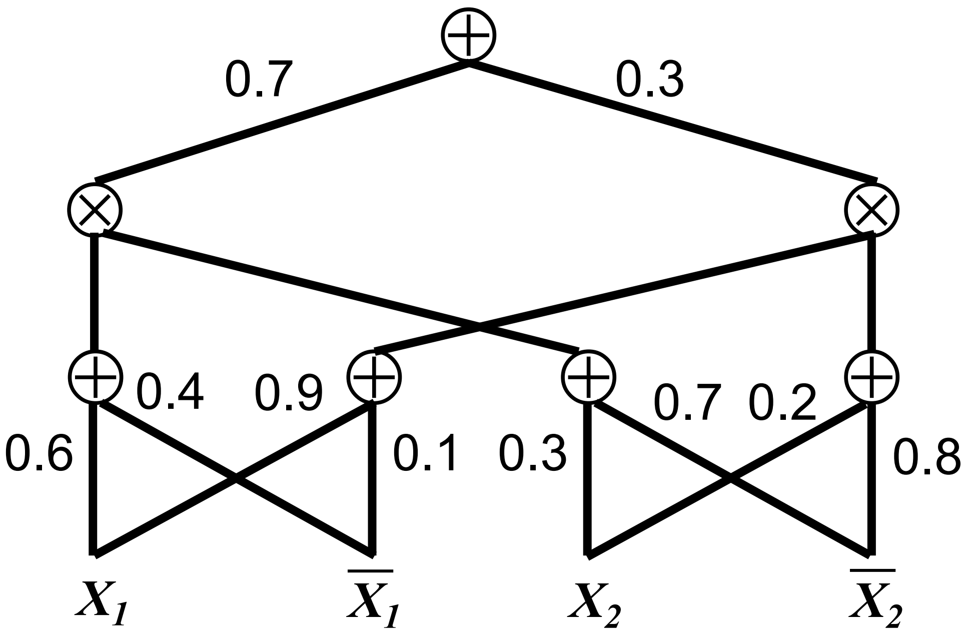
[Peharz, Vergari, Molina, Stelzner, Trapp, Kersting, Ghahramani UAI 2019]

Can we borrow ideas from deep learning for probabilistic graphical models?



Judea Pearl, UCLA
Turing Award 2012

Sum-Product Networks
a deep probabilistic learning framework



Computational graph (kind of TensorFlow graphs) that encodes how to compute probabilities

Inference is linear in size of network

Alternative Representation: Graphical Models as (Deep) Networks

X_1	X_2	$P(X)$
1	1	0.4
1	0	0.2
0	1	0.1
0	0	0.3

$$
\begin{aligned}
P(X) = {} & 0.4 \cdot \mathrm{I}[X_1{=}1] \cdot \mathrm{I}[X_2{=}1] \\
& + 0.2 \cdot \mathrm{I}[X_1{=}1] \cdot \mathrm{I}[X_2{=}0] \\
& + 0.1 \cdot \mathrm{I}[X_1{=}0] \cdot \mathrm{I}[X_2{=}1] \\
& + 0.3 \cdot \mathrm{I}[X_1{=}0] \cdot \mathrm{I}[X_2{=}0]
\end{aligned}
$$

Alternative Representation: Graphical Models as (Deep) Networks

X_1	X_2	$P(X)$
1	**1**	**0.4**
1	0	0.2
0	1	0.1
0	0	0.3

$$P(X) = \mathbf{0.4 \cdot I[X_1=1] \cdot I[X_2=1]} + 0.2 \cdot \mathrm{I}[X_1=1] \cdot \mathrm{I}[X_2=0] + 0.1 \cdot \mathrm{I}[X_1=0] \cdot \mathrm{I}[X_2=1] + 0.3 \cdot \mathrm{I}[X_1=0] \cdot \mathrm{I}[X_2=0]$$

Shorthand using Indicators

X_1	X_2	$P(X)$
1	1	0.4
1	0	0.2
0	1	0.1
0	0	0.3

$$
\begin{aligned}
P(X) = {} & 0.4 \cdot X_1 \cdot X_2 \\
& + 0.2 \cdot X_1 \cdot \overline{X}_2 \\
& + 0.1 \cdot \overline{X}_1 \cdot X_2 \\
& + 0.3 \cdot \overline{X}_1 \cdot \overline{X}_2
\end{aligned}
$$

Summing Out Variables

Let us say, we want to compute $P(X_1 = 1)$

X_1	X_2	$P(X)$
1	**1**	**0.4**
1	**0**	**0.2**
0	1	0.1
0	0	0.3

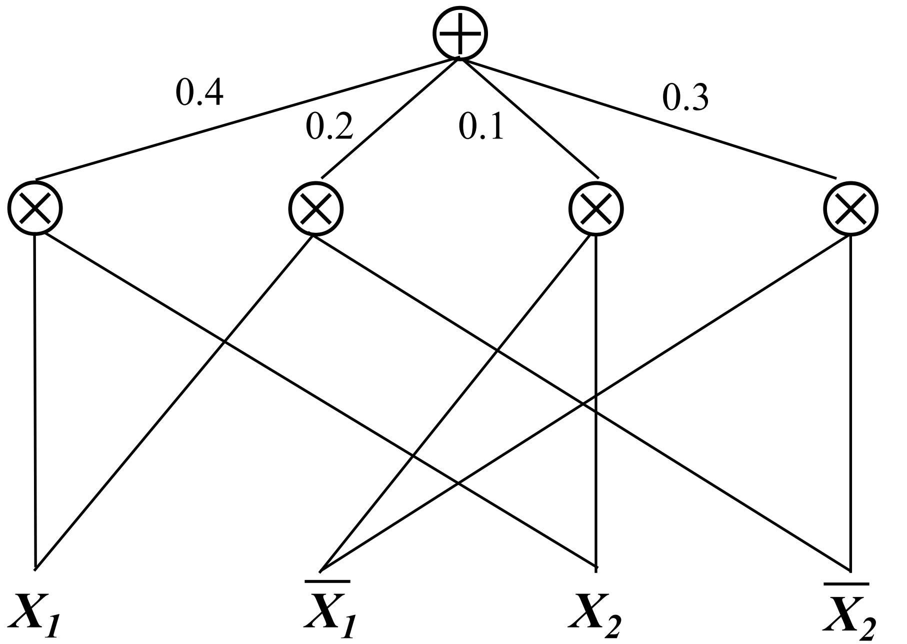
$$P(e) = \mathbf{0.4 \cdot X_1 \cdot X_2} \\ \mathbf{+ 0.2 \cdot X_1 \cdot \overline{X}_2} \\ + 0.1 \cdot \overline{X}_1 \cdot X_2 \\ + 0.3 \cdot \overline{X}_1 \cdot \overline{X}_2$$

Set $X_1 = 1, \overline{X}_1 = 0, X_2 = 1, \overline{X}_2 = 1$

Easy: Set both indicators of X2 to 1

This can be represented as a computational graph

X_1	X_2	$P(X)$
1	1	0.4
1	0	0.2
0	1	0.1
0	0	0.3

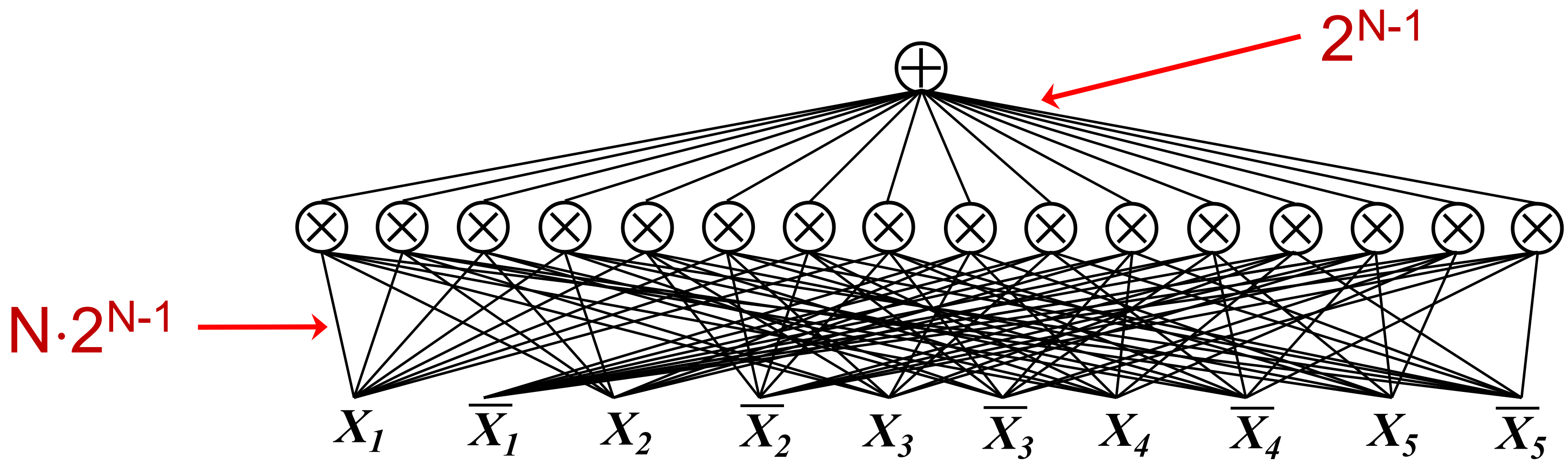


network polynomial

However, the network polynomial of a distribution might be exponentially large

Example: Parity

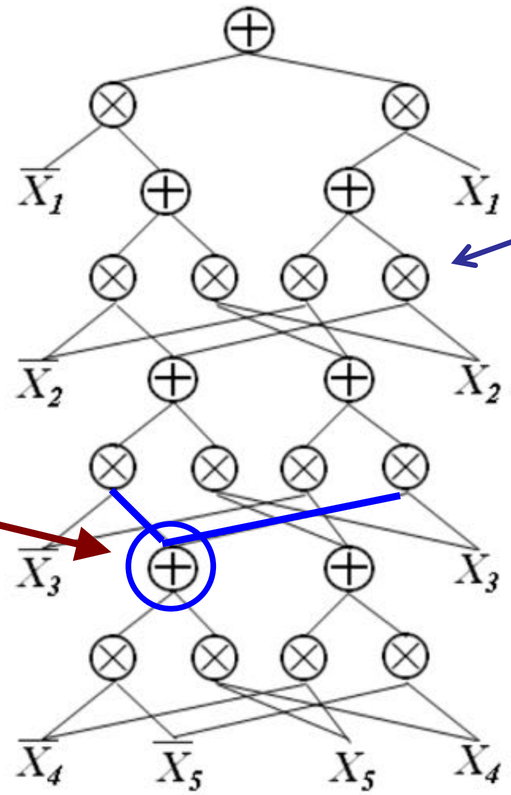
Uniform distribution over states with even number of 1’s



Make the computational graphs deep

Example: Parity

Uniform distribution over states with even number of 1's



Induce many hidden layers

Reuse partial computation

Alternative Representation: Graphical Models as Deep Networks

X_1	X_2	$P(X)$
1	1	0.4
1	0	0.2
0	1	0.1
0	0	0.3

$$
\begin{aligned}
P(X) = {} & 0.4 \cdot \mathrm{I}[X_1{=}1] \cdot \mathrm{I}[X_2{=}1] \\
& + 0.2 \cdot \mathrm{I}[X_1{=}1] \cdot \mathrm{I}[X_2{=}0] \\
& + 0.1 \cdot \mathrm{I}[X_1{=}0] \cdot \mathrm{I}[X_2{=}1] \\
& + 0.3 \cdot \mathrm{I}[X_1{=}0] \cdot \mathrm{I}[X_2{=}0]
\end{aligned}
$$

Alternative Representation: Graphical Models as Deep Networks

X_1	X_2	$P(X)$
1	**1**	**0.4**
1	0	0.2
0	1	0.1
0	0	0.3

$$P(X) = \mathbf{0.4 \cdot I[X_1{=}1] \cdot I[X_2{=}1]}$$
$$+ 0.2 \cdot \mathrm{I}[X_1{=}1] \cdot \mathrm{I}[X_2{=}0]$$
$$+ 0.1 \cdot \mathrm{I}[X_1{=}0] \cdot \mathrm{I}[X_2{=}1]$$
$$+ 0.3 \cdot \mathrm{I}[X_1{=}0] \cdot \mathrm{I}[X_2{=}0]$$

Shorthand for Indicators

X_1	X_2	$P(X)$
1	1	0.4
1	0	0.2
0	1	0.1
0	0	0.3

$$
\begin{aligned}
P(X) = {} & 0.4 \cdot X_1 \cdot X_2 \\
& + 0.2 \cdot X_1 \cdot \overline{X}_2 \\
& + 0.1 \cdot \overline{X}_1 \cdot X_2 \\
& + 0.3 \cdot \overline{X}_1 \cdot \overline{X}_2
\end{aligned}
$$

Sum Out Variables

X_1	X_2	$P(X)$
1	**1**	**0.4**
1	**0**	**0.2**
0	1	0.1
0	0	0.3

$e: X_1 = 1$

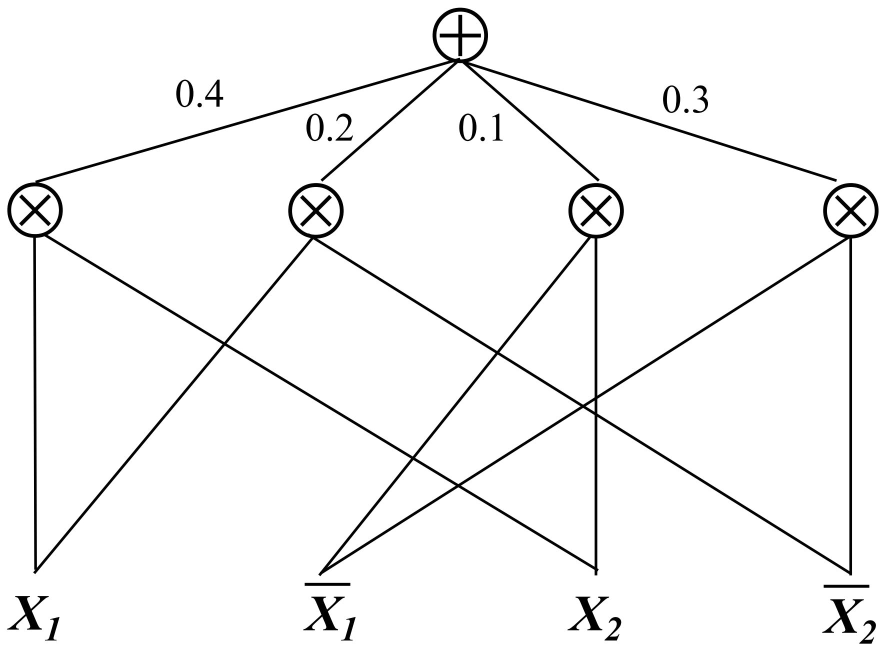
$$
\begin{aligned}
P(e) = &\ \mathbf{0.4 \cdot X_1 \cdot X_2} \\
&+ \mathbf{0.2 \cdot X_1 \cdot \overline{X}_2} \\
&+ 0.1 \cdot \overline{X}_1 \cdot X_2 \\
&+ 0.3 \cdot \overline{X}_1 \cdot \overline{X}_2
\end{aligned}
$$

Set $X_1 = 1, \overline{X}_1 = 0, X_2 = 1, \overline{X}_2 = 1$

Easy: Set both indicators of X2 to 1

Idea: Deeper Network Representation of a Graphical Model that encodes how to compute probabilities

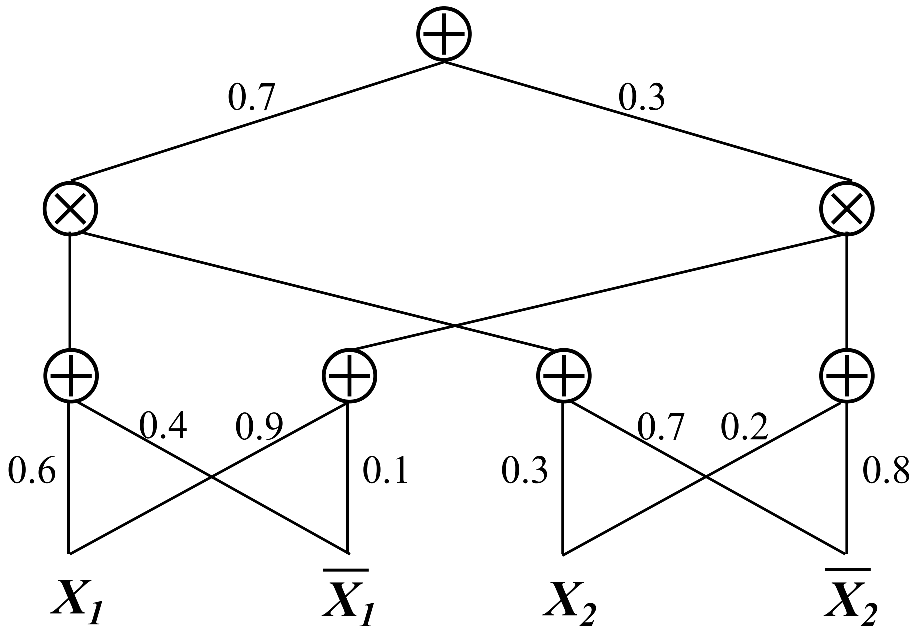
X_1	X_2	$P(X)$
1	1	0.4
1	0	0.2
0	1	0.1
0	0	0.3



Sum-Product Networks* (SPNs)

[Poon, Domingos UAI 2011]

A SPN S is a rooted DAG where:
Nodes: Sum, product, input indicator
Weights on edges from sum to children



*SPNs are an instance of Arithmetic Circuits (ACs). ACs have been introduced into the AI literature more than15 years ago as a tractable representation of probability distributions

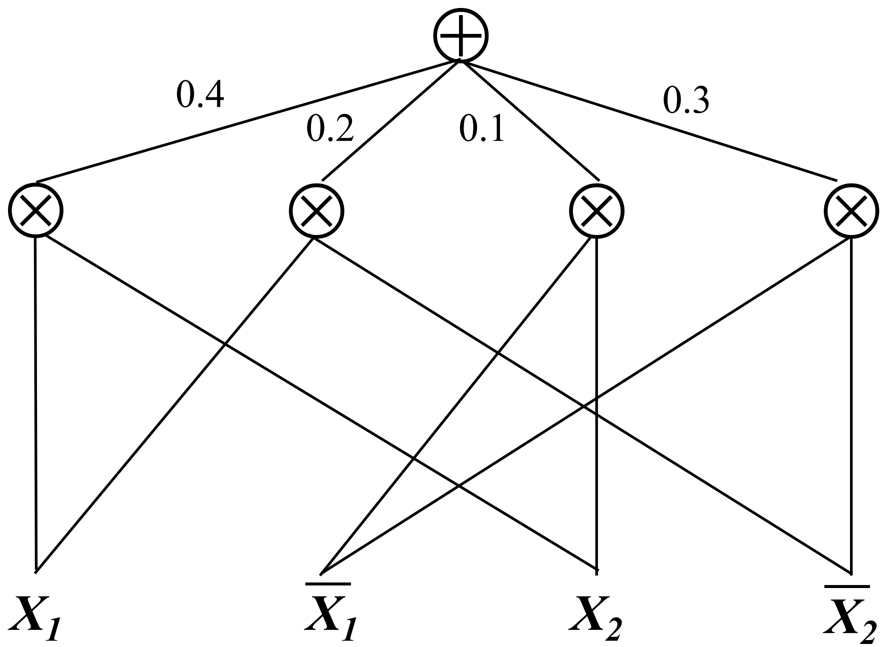
[Darwiche CACM 48(4):608-647 2001]

Your turn!

$$P\left(x \mid y\right) = \frac{P(x, y)}{P(y)}$$

What is P(X_2)? What is P(X_1|X_2=1)?

X_1	X_2	$P(X)$
1	1	0.4
1	0	0.2
0	1	0.1
0	0	0.3



You have 10 minutes!

[Poon, Domingos UAI’11; Molina, Natarajan, Kersting AAAI’17; Vergari, Peharz, Di Mauro, Molina, Kersting, Esposito AAAI ’18; Molina, Vergari, Di Mauro, Esposito, Natarajan, Kersting AAAI ‘18]

SPFlow: An Easy and Extensible Library for Sum-Product Networks

[Molina, Vergari, Stelzner, Peharz, Subramani, Poupart, Di Mauro, Kersting 2019]

TECHNISCHE UNIVERSITÄT DARMSTADT · UNIVERSITÀ DEGLI STUDI DI BARI ALDO MORO · UNIVERSITY OF WATERLOO · Max Planck Institute for Intelligent Systems · UNIVERSITY OF CAMBRIDGE · VECTOR INSTITUTE · CAML · MADESI · DFG · Federal Ministry of Education and Research

195 commits · 2 branches · 0 releases · 6 contrib...

Branch: master · New pull request · Create new file · Upload files · Find file · Clone or download

https://github.com/SPFlow/SPFlow

```
from spn.structure.leaves.parametric.Parametric import Categorical

from spn.structure.Base import Sum, Product

from spn.structure.base import assign_ids, rebuild_scopes_bottom_up


p0 = Product(children=[Categorical(p=[0.3, 0.7], scope=1), Categorical(p=[0.4, 0.6], scope=2)])
p1 = Product(children=[Categorical(p=[0.5, 0.5], scope=1), Categorical(p=[0.6, 0.4], scope=2)])
s1 = Sum(weights=[0.3, 0.7], children=[p0, p1])
p2 = Product(children=[Categorical(p=[0.2, 0.8], scope=0), s1])
p3 = Product(children=[Categorical(p=[0.2, 0.8], scope=0), Categorical(p=[0.3, 0.7], scope=1)])
p4 = Product(children=[p3, Categorical(p=[0.4, 0.6], scope=2)])
spn = Sum(weights=[0.4, 0.6], children=[p2, p4])

assign_ids(spn)
rebuild_scopes_bottom_up(spn)

return spn
```

Domain Specific Language, Inference, EM, and Model Selection as well as Compilation of SPNs into TF and PyTorch and also into flat, library-free code even suitable for running on devices: C/C++,GPU, FPGA

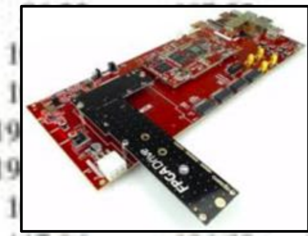
SPFlow, an open-source Python library providing a simple interface to inference, learning and manipulation routines for deep and tractable probabilistic models called Sum-Product Networks (SPNs). The library allows one to quickly create SPNs both from data and through a domain specific language (DSL). It efficiently implements several probabilistic inference

[Sommer, Oppermann, Molina, Binnig, Kersting, Koch ICDD 2018]

TABLE II

PERFORMANCE COMPARISON. BEST END-TO-END THROUGHPUTS (T), EXCLUDING THE CYCLE COUNTER MEASUREMENTS, ARE DENOTED BOLD.

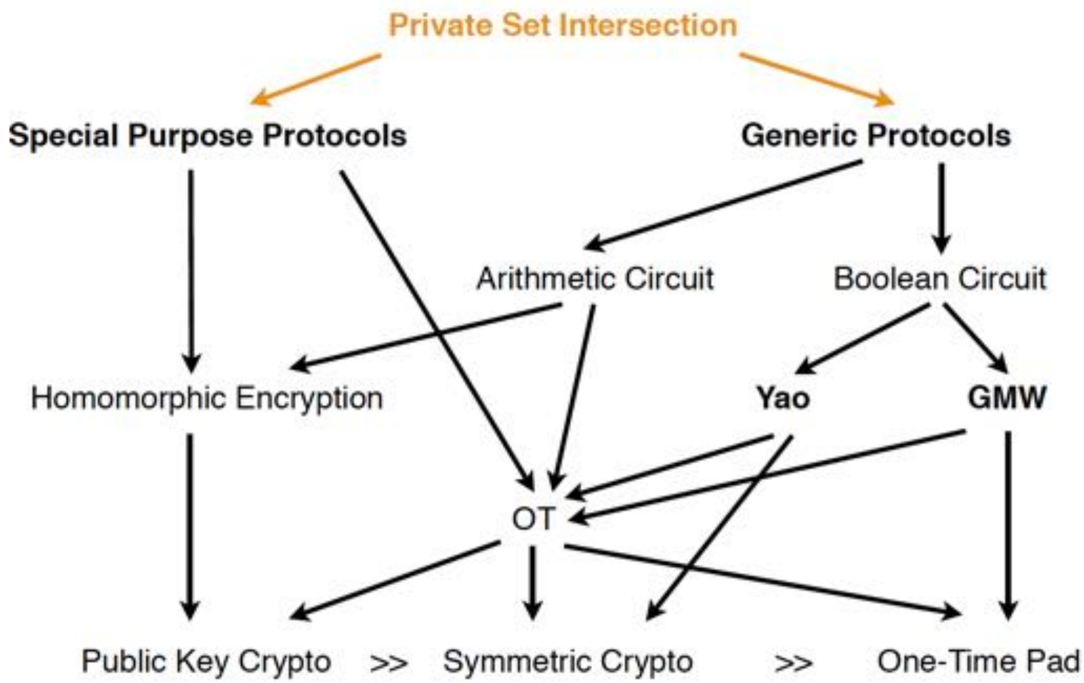
Dataset	Rows	CPU (μs)	T-CPU (rows/ μs)	CPUF (μs)	T-CPUF (rows/ μs)	GPU (μs)	T-GPU (rows/ μs)	FPGA Cycle Counter	FPGAC (μs)	T-FPGAC (rows/ μs)	FPGA (μs)	T-FPGA (rows/ μs)
Accidents	17009	2798.27	[illegible]	[illegible]	7.87	63090.94	0.27	17249	[illegible]	[illegible]	[illegible]	**24.44**
Audio	20000	4271.78	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	20317	[illegible]	[illegible]	[illegible]	**26.28**
Netflix	20000	4892.22	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	20322	[illegible]	[illegible]	[illegible]	**30.58**
MSNBC200	388434	15476.05	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	388900	[illegible]	[illegible]	[illegible]	**77.56**
MSNBC300	388434	10060.78	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	388810	[illegible]	[illegible]	[illegible]	**78.74**
NLTCS	21574	791.80	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	21904	[illegible]	[illegible]	[illegible]	**38.12**
Plants	23215	3621.71	6.41	3521.04	6.59	67004.41	0.35	23592	117.96	196.80	778.00	**29.84**
NIPS5	10000	25.11	**398.31**	26.37	379.23	8210.32	1.22	10236	51.18	195.39	337.30	[illegible]
NIPS10	10000	83.60	**119.61**	84.39	118.49	11550.82	0.87	10279	51.40	194.57	464.30	21.54
NIPS20	10000	191.30	52.27	182.73	**54.72**	18689.04	0.54	10285	51.43	194.46	543.60	18.40
NIPS30	10000	387.61	25.80	349.84	**28.58**	25355.93	0.39	10308	51.80	193.06	592.30	16.88
NIPS40	10000	551.64	18.13	471.26	**21.22**	30820.49	0.32	10306	51.53	194.06	632.20	15.82
NIPS50	10000	812.44	12.31	792.13	12.62	36355.60	0.28	10559	52.80	189.41	720.60	**13.88**
NIPS60	10000	1046.38	9.56	662.53	**15.09**	40778.36	0.25	12271	61.36	162.99	799.20	12.51
NIPS70	10000	1148.17	8.71	1134.80	8.81	46759.26	0.21	14022	70.11	142.63	858.60	**11.65**
NIPS80	10000	1556.99	6.42	1277.81	7.83	63217.99	0.16	14275	78.51	127.37	961.80	**10.40**



How do we do deep learning offshore?

MADESI

Federal Ministry of Education and Research



There are generic protocols to validate computations on authenticated data without knowledge of the secret key

DNA MSPN
Gates: 298208 Yao Bytes: 9542656 Depth: 615

DNA PSPN
Gates: 228272 Yao Bytes: 7304704 Depth: 589

NIPS MSPN
Gates: 1001477 Yao Bytes: 32047264 Depth: 970

Homomorphic sum-product network

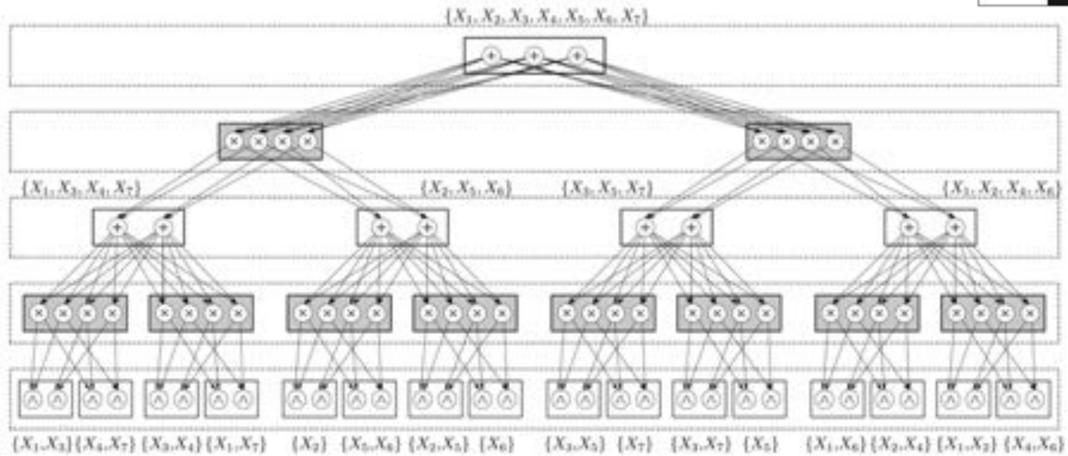
[Molina, Weinert, Treiber, Schneider, Kersting to be submitted 2019]

Random sum-product networks

[Peharz, Vergari, Molina, Stelzner, Trapp, Kersting, Ghahramani UAI 2019]

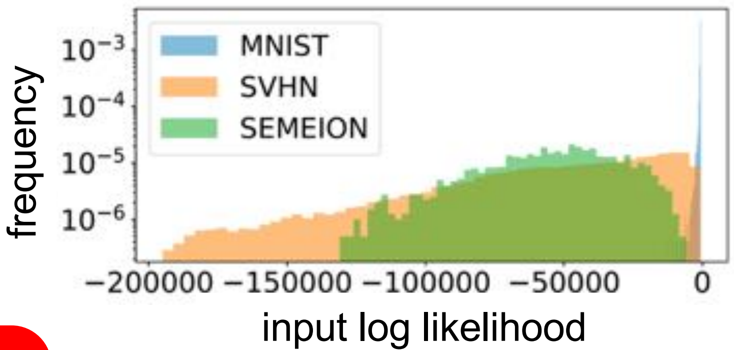
UNIVERSITY OF CAMBRIDGE
Max Planck Institute for Intelligent Systems
TECHNISCHE UNIVERSITÄT DARMSTADT
UNI GRAZ
UBER AI Labs

Conference on Uncertainty in Artificial Intelligence
Tel Aviv, Israel
July 22 - 25, 2019
uai2019



Build a random SPN structure. This can be done in an informed way or completely at random

		RAT-SPN	MLP	vMLP
Accuracy	MNIST	98.19 (8.5M)	98.32 (2.64M)	98.09 (5.28M)
	F-MNIST	89.52 (0.65M)	90.81 (9.28M)	89.81 (1.07M)
	20-NG	47.8 (0.37M)	49.05 (0.31M)	48.81 (0.16M)
Cross-Entropy	MNIST	0.0852 (17M)	0.0874 (0.82M)	0.0974 (0.22M)
	F-MNIST	0.3525 (0.65M)	0.2965 (0.82M)	0.325 (0.29M)
	20-NG	1.6954 (1.63M)	1.6180 (0.22M)	1.6263 (0.22M)



outliers
prototypes
outliers
prototypes

SPNs can have similar predictive performances as (simple) DNNs

SPNs can distinguish the datasets

SPNs know when they do not know by design

Your turn!

Mission completed? Just give me data and everything is done by ML/AI?

You have 5 minutes!

Reproducibility Crisis in Science (2016)

M. Baker: „1,500 scientists lift the lid on reproducibility". Nature, 2016 May 26;533(7604):452-4. doi: 10.1038/533452
https://www.nature.com/news/1-500-scientists-lift-the-lid-on-reproducibility-1.19970?proof=true

The New York Times

Opinion

A.I. Is Harder Than You Think

By Gary Marcus and Ernest Davis

Mr. Marcus is a professor of psychology and neural science. Mr. Davis is a professor of computer science.

May 18, 2018

Reproducibility Crisis in ML & AI (2018)

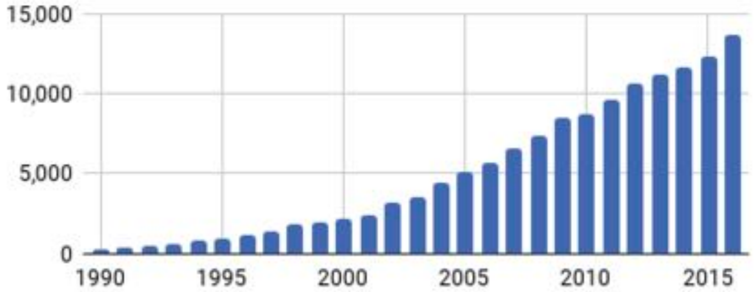


Figure 1: Growth of published reinforcement learning papers. Shown are the number of RL-related publications (y-axis) per year (x-axis) scraped from Google Scholar searches.

Joelle Pineau

McGill

Facebook AI Research (FAIR)

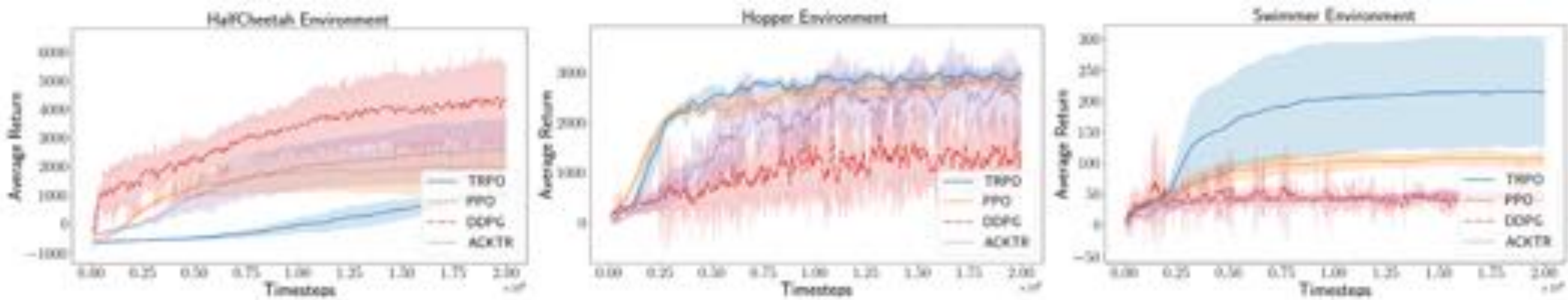
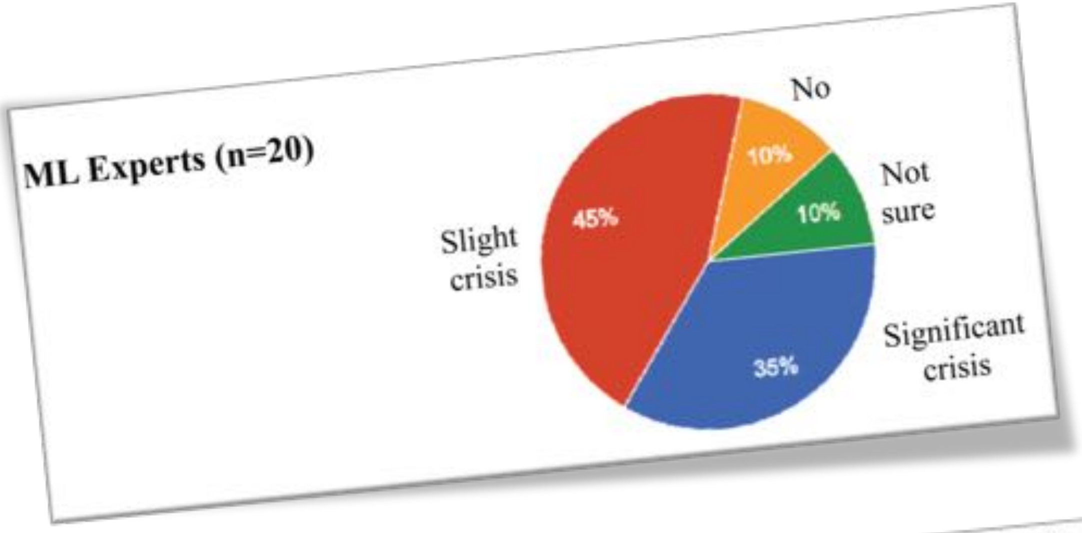
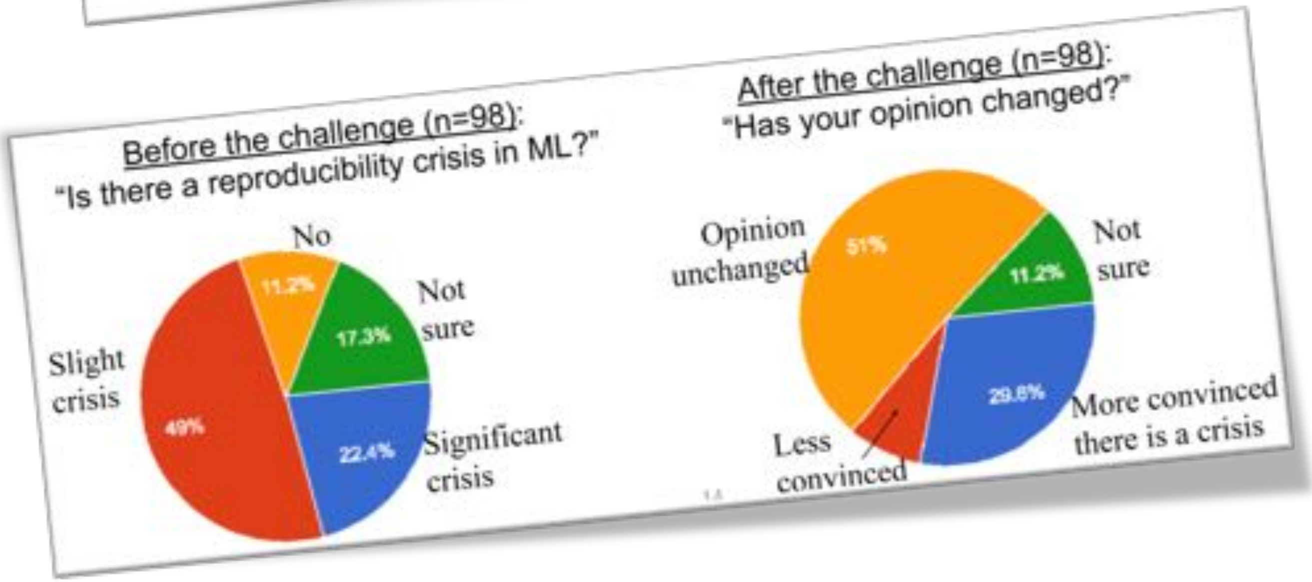


Figure 4: Performance of several policy gradient algorithms across benchmark MuJoCo environment suites

P. Henderson et al.: “Deep Reinforcement learning that Matters“. AAAI 2018

Reproducibility Crisis in ML & AI (2018)





J. Pineau: „The ICLR 2018 Reproducibility Challenge".
Talk at the MLTRAIN@RML Workshop at ICML 2018

Joelle Pineau

McGill

Facebook AI Research (FAIR)

Survey participants:

- 54 challenge participants
- 30 authors of ICLR submissions targeted by reproducibility effort
- 14 others (random volunteers, other ICLR authors, ICLR area chair & reviewers, course instructors)

MLTRAIN

Nikolaos Vasiloglou

ismion

NIPS HIGHLIGHTS, LEARN HOW TO CODE A PAPER WITH STATE OF THE ART FRAMEWORKS

Dec 09 @ 08:50 AM - 06:05 PM   NIPS, Los Angeles, California

ENABLING REPRODUCIBILITY IN MACHINE LEARNING MLTRAIN@RML (ICML 2018)

Jul 14 @ 08:30 AM - 06:00 PM   Stockholmsmässan

Mila

Yoshua Bengio (Turing Award 2019)

frontiers in Big Data | Machine Learning and Artificial Intelligence

First Machine Learning and Artificial Intelligence journal that explicitely welcomes replication studies and code review papers

Sriraam Natarajan

UT DALLAS
The University of Texas at Dallas

A lot of systems to support reproducible ML research

OpenML beta_2

Machine learning, better, together

Joaquin Vanschoren

Technische Universiteit Eindhoven University of Technology

20328 data sets — Find or add **data** to analyse

68724 tasks — Download or create scientific **tasks**

6994 flows — Find or add data analysis **flows**

9749541 runs — Upload and explore all **results** online.

Percy Lang

Stanford University

CodaLab

Accelerating reproducible computational research.

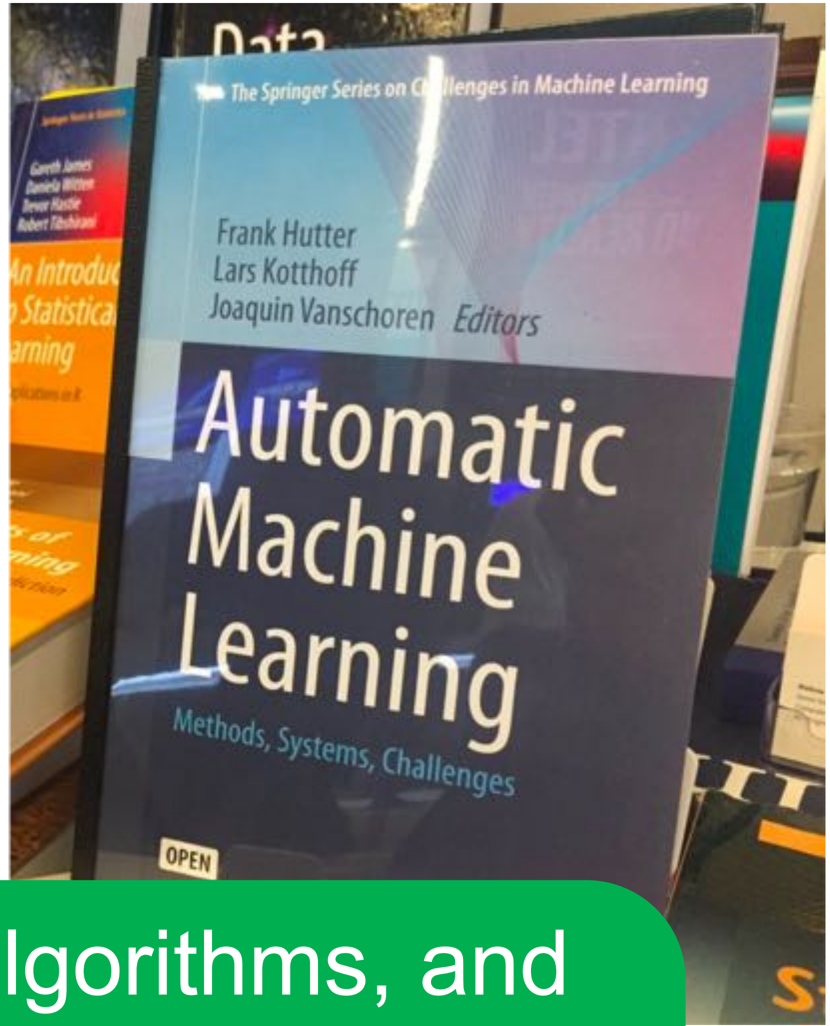
Worksheets

Run reproducible experiments and create executable papers using worksheets.

Competitions

Enter an existing competition to solve challenging data problems, or host your own.

However, there are not enough data scientists, statisticians, machine learning and AI experts

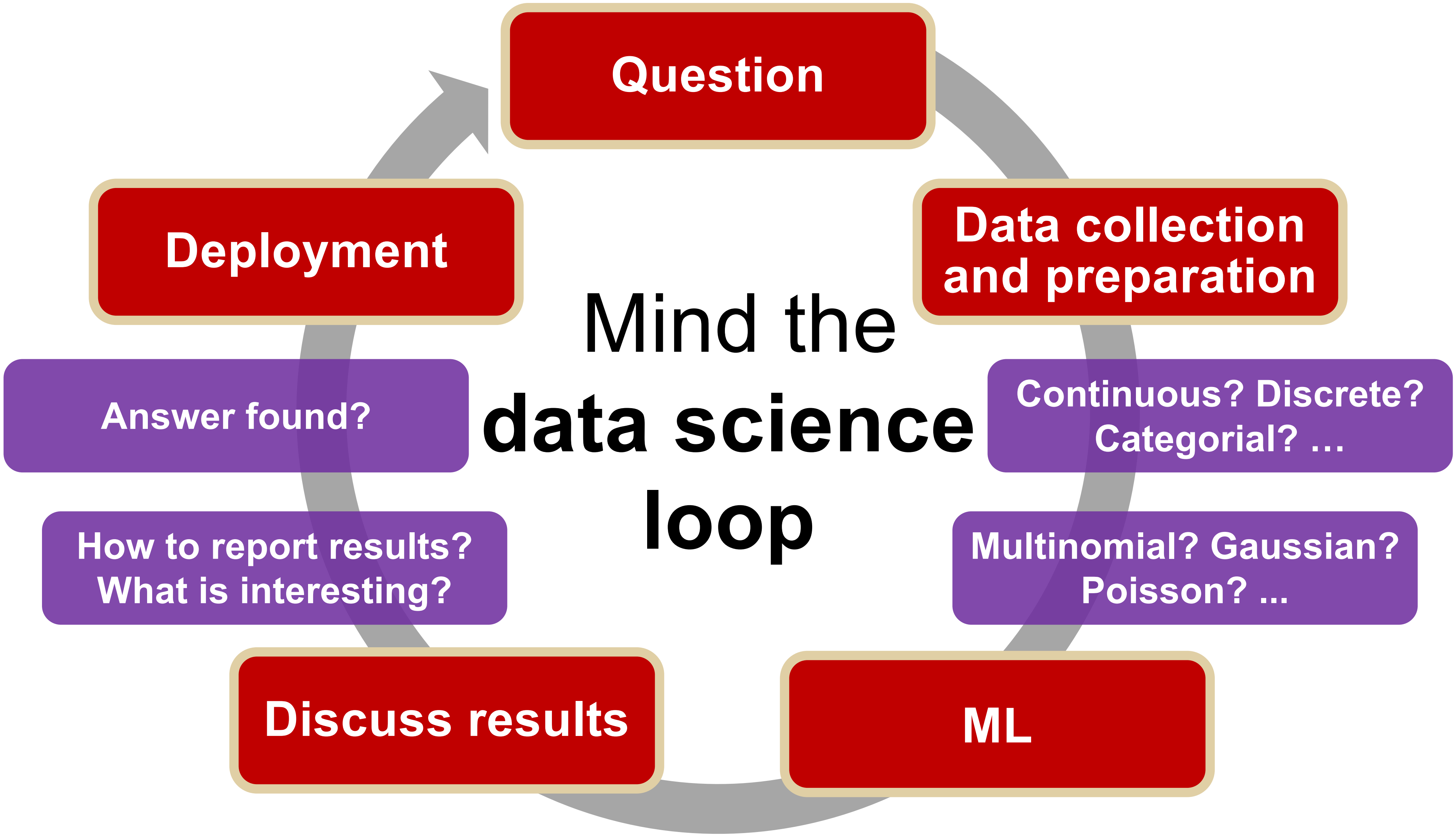


Provide the foundations, algorithms, and tools to develop systems that ease and support building ML/AI models as much as possible and in turn help reproducing and hopfeully even justifying our results

Your turn!

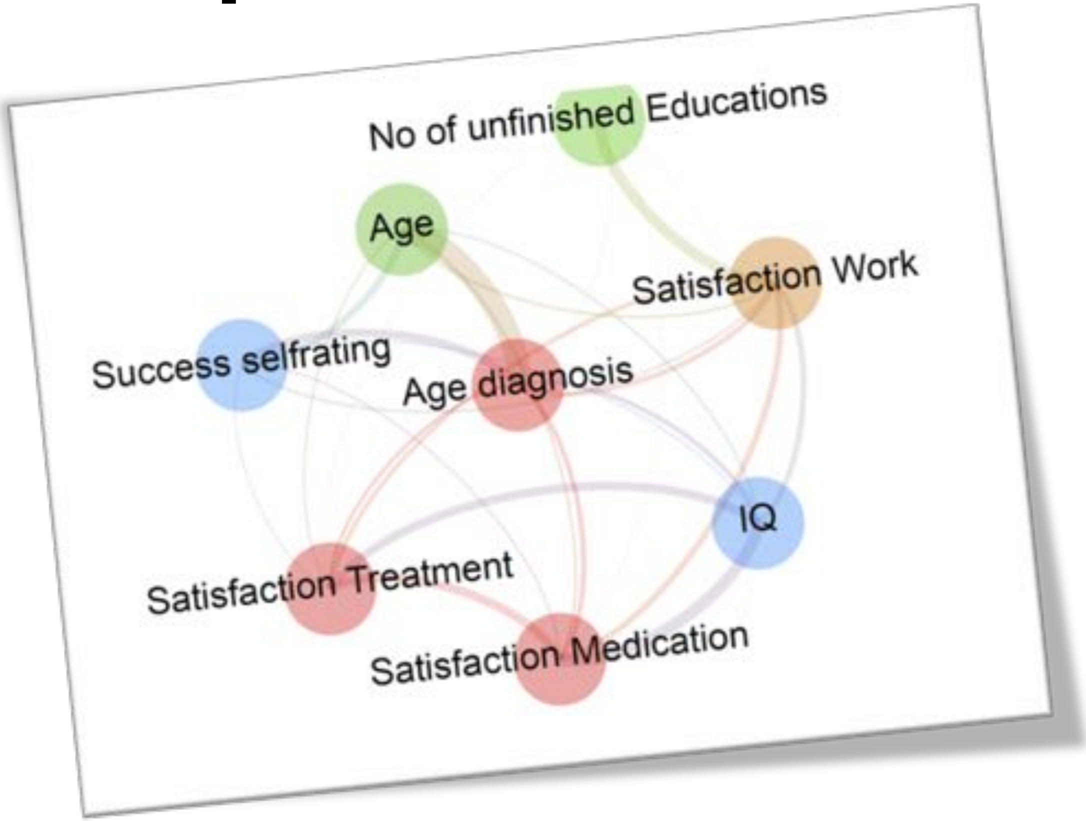
Do you think AutoML is solving everything?

You have 5 minutes!

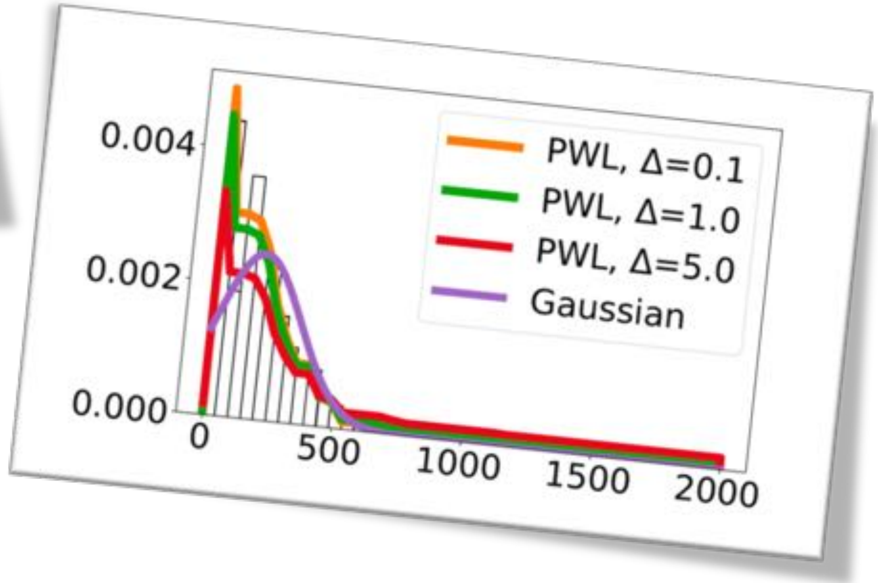

Question
Data collection and preparation
Continuous? Discrete? Categorial? …
Multinomial? Gaussian? Poisson? …
ML
Discuss results
How to report results? What is interesting?
Answer found?
Deployment
Mind the data science loop

[Molina, Natarajan, Vergari, Di Mauro, Esposito, Kersting AAAI 2018]

Distribution-agnostic Deep Probabilistic Learning



Use nonparametric independency tests and piece-wise linear approximations



[Molina, Natarajan, Vergari, Di Mauro, Esposito, Kersting AAAI 2018]

Distribution-agnostic Deep Probabilistic Learning

Use nonparametric independency tests and piece-wise linear approximations

However, we have to provide the statistical types and do not gain insights into the parametric forms of the variables. **Are they Gaussians? Gammas? …**

[Vergari, Molina, Peharz, Ghahramani, Kersting, Valera AAAI 2019]

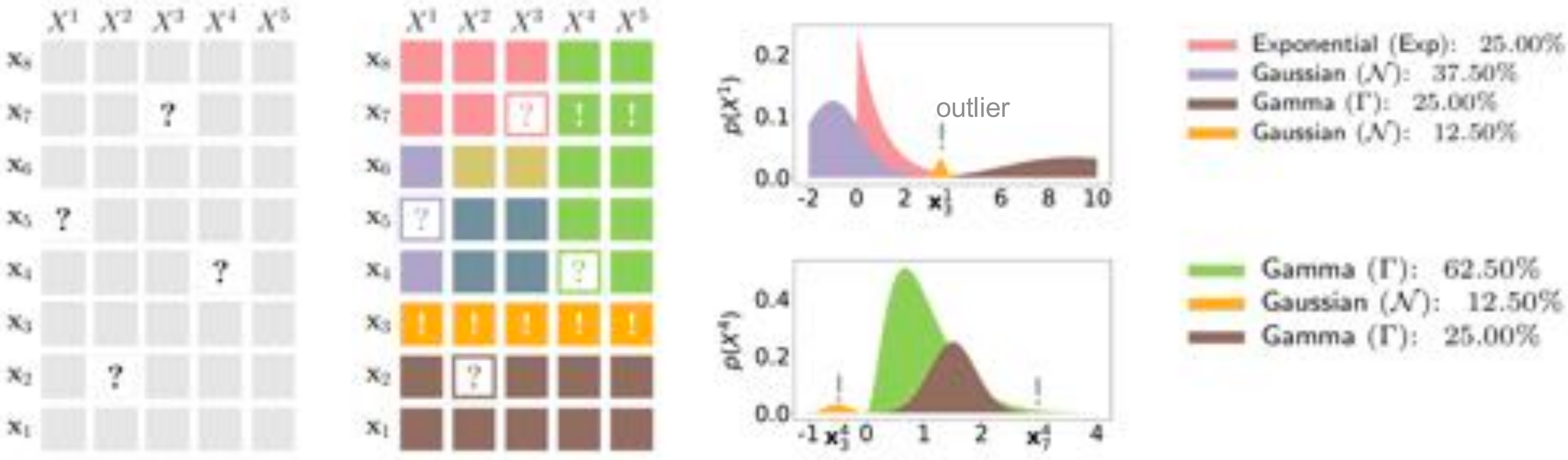
CAML DFG MADESI Federal Ministry of Education and Research

The Automatic Data Scientist

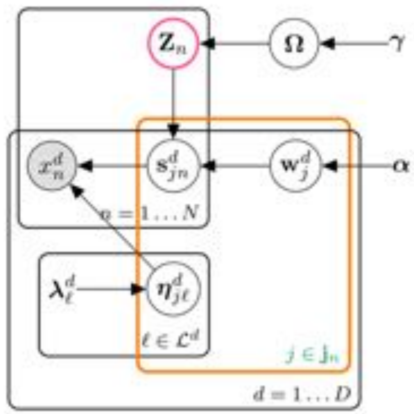
UBER AI Labs · UNIVERSITY OF CAMBRIDGE · Max Planck Institute for Intelligent Systems · TECHNISCHE UNIVERSITÄT DARMSTADT

AAAI-19: Thirty-Third AAAI Conference on Artificial Intelligence
January 27 – February 1, 2019, Hilton Hawaiian Village, Honolulu, Hawaii, USA

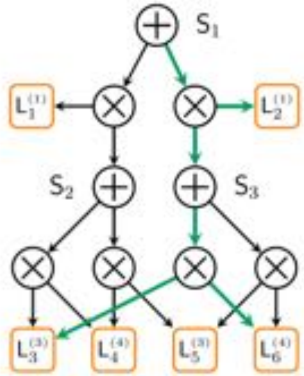
missing value



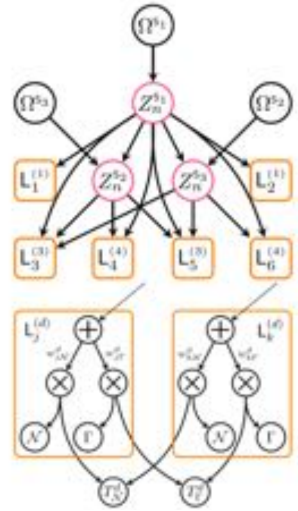
We can even automatically discovers the statistical types and parametric forms of the variables



Bayesian Type Discovery



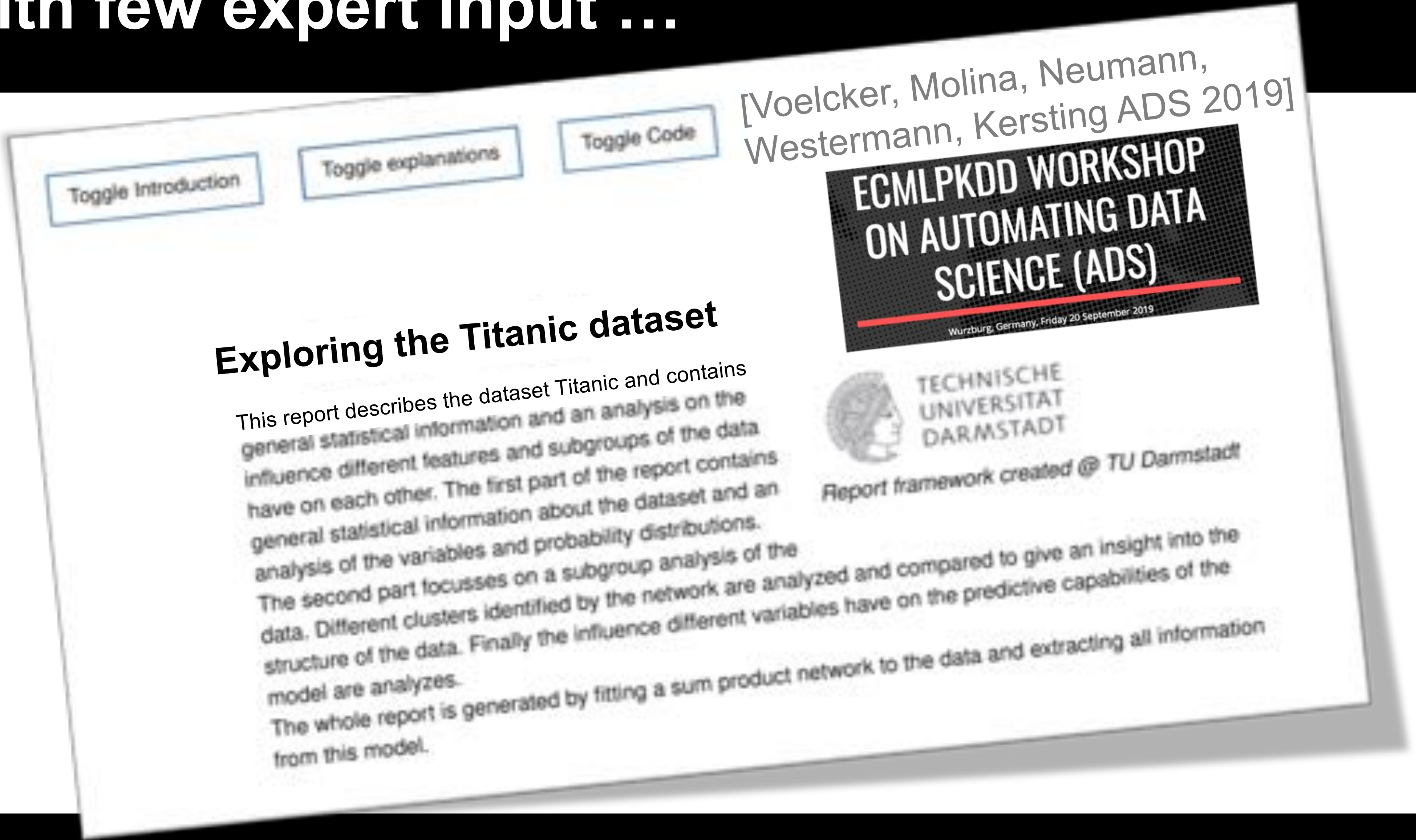
Mixed Sum-Product Network



Automatic Statistician

That is, the machine understands the data with few expert input …

[Voelcker, Molina, Neumann, Westermann, Kersting ADS 2019]



…and can compile data reports automatically

Your turn!

But now we have completed our mission! Really

P(heart attack | )?

P(heart attack | )?


The New York Times

Opinion

A.I. Is Harder Than You Think

By Gary Marcus and Ernest Davis

Mr. Marcus is a professor of psychology and neural science. Mr. Davis is a professor of computer science.

May 18, 2018

P(heart attack | )?

The New York Times

Opinion

A.I. Is Harder Than You Think

and Data Science

By Gary Marcus and Ernest Davis

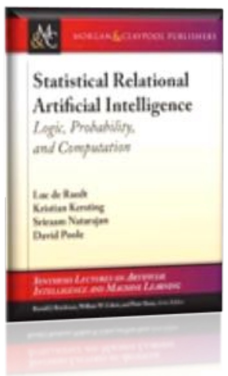
Mr. Marcus is a professor of psychology and neural science. Mr. Davis is a professor of computer science.

May 18, 2018

P(heart attack | )?

Crossover of ML and DS with data & programming abstractions

De Raedt, Kersting, Natarajan, Poole: Statistical Relational Artificial Intelligence: Logic, Probability, and Computation. Morgan and Claypool Publishers, ISBN: 9781627058414, 2016.



KATHOLIEKE UNIVERSITEIT LEUVEN

THE UNIVERSITY OF TEXAS AT DALLAS

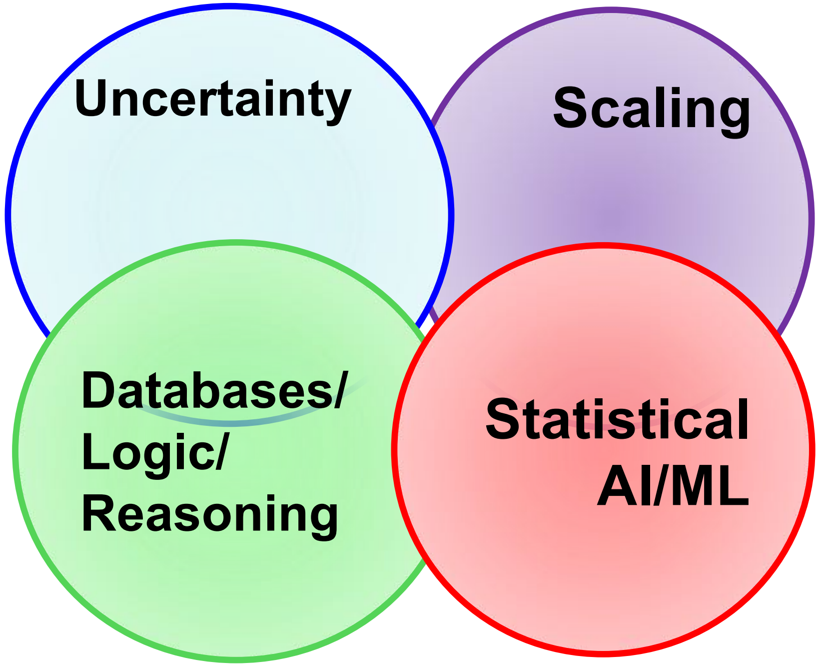
TECHNISCHE UNIVERSITÄT DARMSTADT

THE UNIVERSITY OF BRITISH COLUMBIA

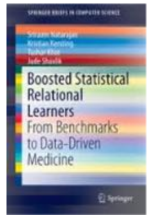
building general-purpose data science and ML machines

make the ML/DS expert more effective

increases the number of people who can successfully build ML/DS applications



Natarajan, Khot, Kersting, Shavlik. Boosted Statistical Relational Learners. Springer Brief 2015

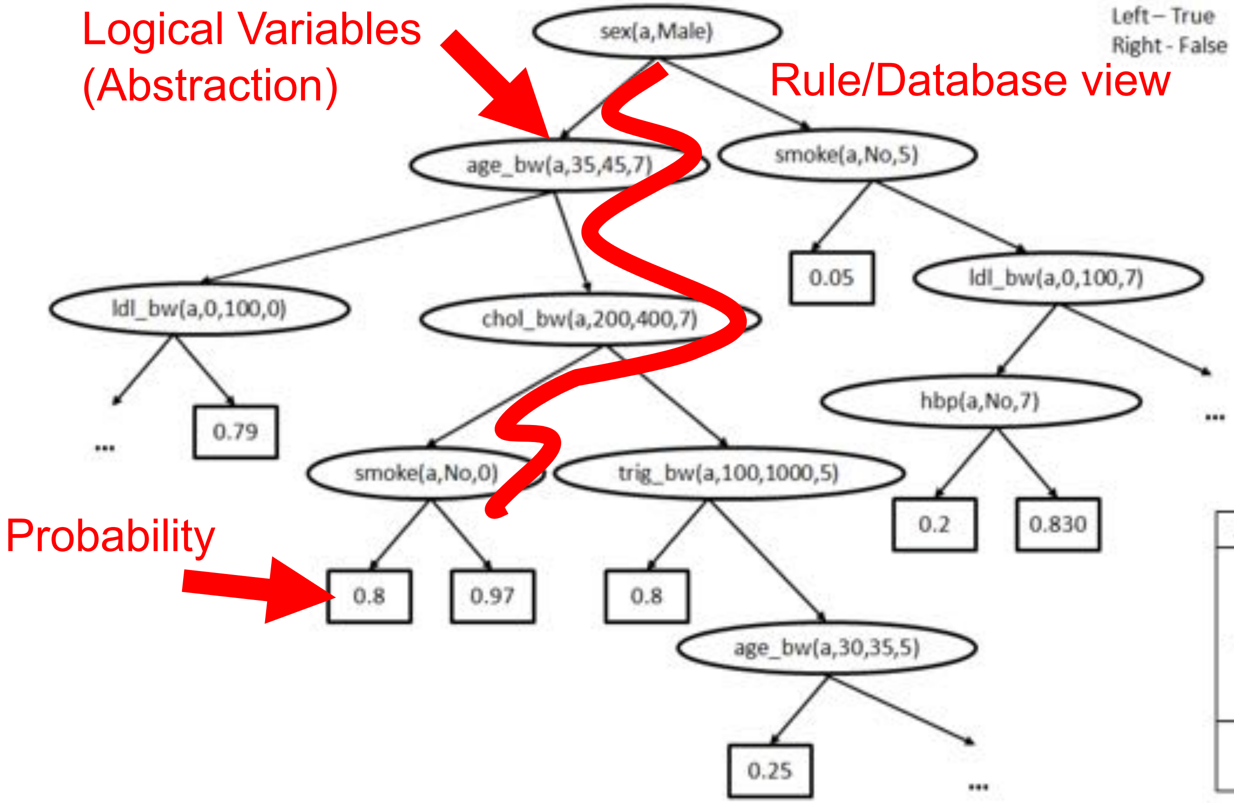


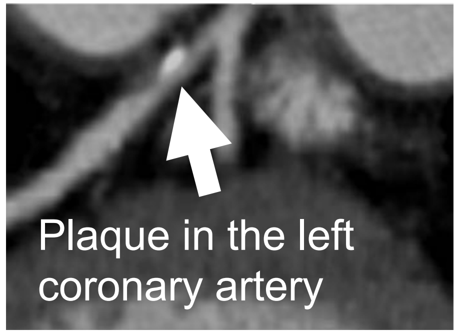
Understanding Electronic Health Records

Atherosclerosis is the cause of the majority of Acute Myocardial Infarctions (heart attacks)

TECHNISCHE UNIVERSITÄT DARMSTADT

THE UNIVERSITY OF TEXAS AT DALLAS





[Circulation; 92(8), 2157-62, 1995; JACC; 43, 842-7, 2004]

Algorithm	Accuracy	AUC-ROC
J48	0.667	0.607
SVM	0.667	0.5
AdaBoost	0.667	0.608
Bagging	0.677	0.613
NB	0.75	0.653
RPT	0.669*	0.778
RFGB	0.667*	0.819

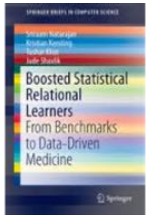
The higher, the better

25%

Algorithm for Mining Markov Logic Networks	Likelihood The higher, the better	AUC-ROC The higher, the better	AUC-PR The higher, the better	Time The lower, the better
Boosting	0.81	0.96	0.93	**9s**
LSM	0.73	0.54	0.62	**93 hrs**

11% 78% 50% 37200x faster

[Kersting, Driessens ICML´08; Karwath, Kersting, Landwehr ICDM´08; Natarajan, Joshi, Tadepelli, Kersting, Shavlik. IJCAI´11; Natarajan, Kersting, Ip, Jacobs, Carr IAAI `13; Yang, Kersting, Terry, Carr, Natarajan AIME ´15; Khot, Natarajan, Kersting, Shavlik ICDM´13, MLJ´12, MLJ´15, Yang, Kersting, Natarajan BIBM`17]



Natarajan, Khot, Kersting, Shavlik. Boosted Statistical Relational Learners. Springer Brief 2015

TECHNISCHE UNIVERSITÄT DARMSTADT

UTD THE UNIVERSITY OF TEXAS AT DALLAS

https://starling.utdallas.edu/software/boostsrl/wiki/

StARLinG LAB — People | Publications | Projects | Software | Datasets | Blog

BOOSTSRL BASICS

- Getting Started
- File Structure
- Basic Parameters
- Advanced Parameters
- Basic Modes
- Advanced Modes

ADVANCED BOOSTSRL

- Default (RDN-Boost)
- MLN-Boost
- Regression
- One-Class Classification
- Cost-Sensitive SRL
- Learning with Advice
- Approximate Counting
- Discretization of Continuous-Valued Attributes
- Lifted Relational Random Walks
- Grounded Relational Random Walks

APPLICATIONS

- Natural Language Processing

BoostSRL Wiki

BoostSRL (Boosting for Statistical Relational Learning) is a gradient-boosting based approach to learning different types of SRL models. As with the standard gradient-boosting approach, our approach turns the model learning problem to learning a sequence of regression models. The key difference to the standard approaches is that we learn relational regression models i.e., regression models that operate on relational data. We assume the data in a predicate logic format and the output are essentially first-order regression trees where the inner nodes contain conjunctions of logical predicates. For more details on the models and the algorithm, we refer to our book on this topic.

Sriraam Natarajan, Tushar Khot, Kristian Kersting and Jude Shavlik, Boosted Statistical Relational Learners: From Benchmarks to Data-Driven Medicine . SpringerBriefs in Computer Science, ISBN: 978-3-319-13643-1, 2015

Human-in-the-loop learning

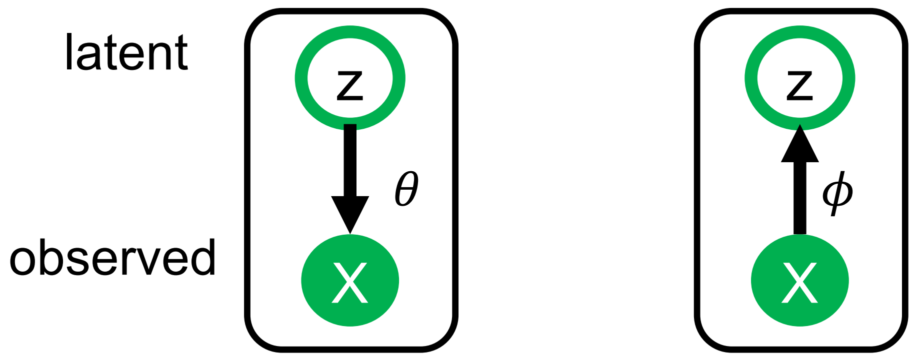
In general, computing the exact posterior is intractable, i.e., inverting the generative process to determine the state of latent variables corresponding to an input is time-consuming and error-prone.

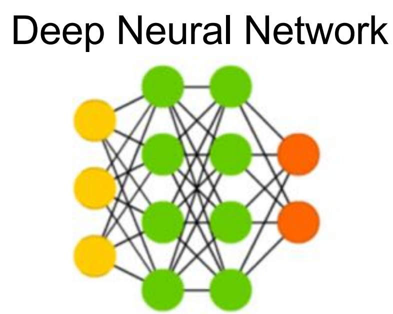
Deep Probabilistic Programming

```python
import pyro.distributions as dist

def model(data):
    # define the hyperparameters that control the beta prior
    alpha0 = torch.tensor(10.0)
    beta0 = torch.tensor(10.0)
    # sample f from the beta prior
    f = pyro.sample("latent_fairness", dist.Beta(alpha0, beta0))
    # loop over the observed data
    for i in range(len(data)):
        # observe datapoint i using the bernoulli
        # likelihood Bernoulli(f)
        pyro.sample("obs_{}".format(i), dist.Bernoulli(f), obs=data[i])
```

```python
def guide(data):
    # register the two variational parameters with Pyro.
    alpha_q = pyro.param("alpha_q", torch.tensor(15.0),
                         constraint=constraints.positive)
    beta_q = pyro.param("beta_q", torch.tensor(15.0),
                        constraint=constraints.positive)
    # sample latent_fairness from the distribution Beta(alpha_q, beta_q)
    pyro.sample("latent_fairness", dist.Beta(alpha_q, beta_q))
```

(2) Ease the implementation by some high-level, probabilistic programming language





(1) Instead of optimizating variational parameters for every new data point, use a deep network to predict the posterior given X [Kingma, Welling 2013, Rezende et al. 2014]

UBER AI Labs · UNIVERSITY OF CAMBRIDGE · Max Planck Institute for Intelligent Systems · UNI GRAZ · TECHNISCHE UNIVERSITÄT DARMSTADT

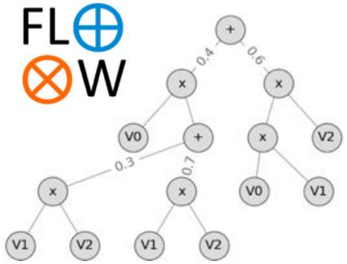
[Stelzner, Molina, Peharz, Vergari, Trapp, Valera, Ghahramani, Kersting ProgProb 2018]

Sum-Product Probabilistic Programming

```
import pyro.distributions as dist

def model(data):
    # define the hyperparameters that control the beta prior
    alpha0 = torch.tensor(10.0)
    beta0 = torch.tensor(10.0)
    # sample f from the beta prior
    f = pyro.sample("latent_fairness", dist.Beta(alpha0, beta0))
    # loop over the observed data
    for i in range(len(data)):
        # observe datapoint i using the bernoulli
        # likelihood Bernoulli(f)
        pyro.sample("obs_{}".format(i), dist.Bernoulli(f), obs=data[i])
```

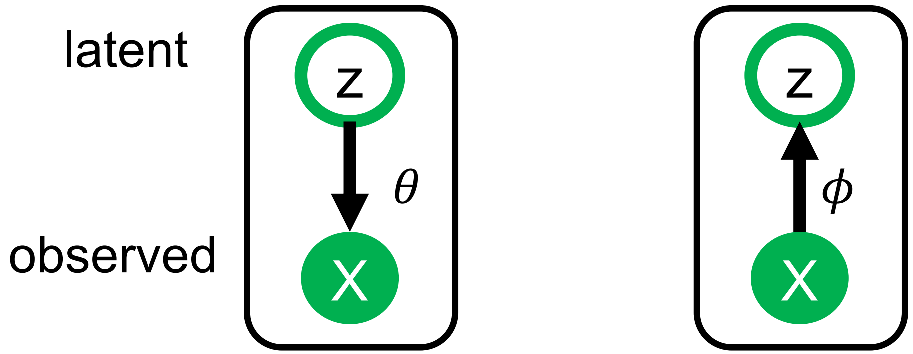
```
def guide(data):
    # register the two variational parameters with Pyro.
    alpha_q = pyro.param("alpha_q", torch.tensor(15.0),
                         constraint=constraints.positive)
    beta_q = pyro.param("beta_q", torch.tensor(15.0),
                        constraint=constraints.positive)
    # sample latent_fairness from the distribution Beta(alpha_q, beta_q)
    pyro.sample("latent_fairness", dist.Beta(alpha_q, beta_q))
```

Sum-Product Network



(2) Ease the implementation by some high-level, probabilistic programming language

Deep Neural Network



(1) Instead of optimizating variational parameters for every new data point, use a deep network to predict the posterior given X [Kingma, Welling 2013, Rezende et al. 2014]

Unsupervised scene understanding

[Stelzner, Peharz, Kersting ICML 2019]

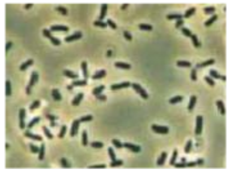
UNIVERSITY OF CAMBRIDGE

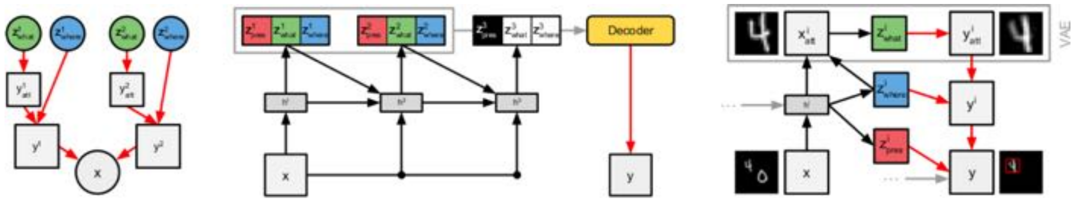
TECHNISCHE UNIVERSITÄT DARMSTADT

ICML | 2019

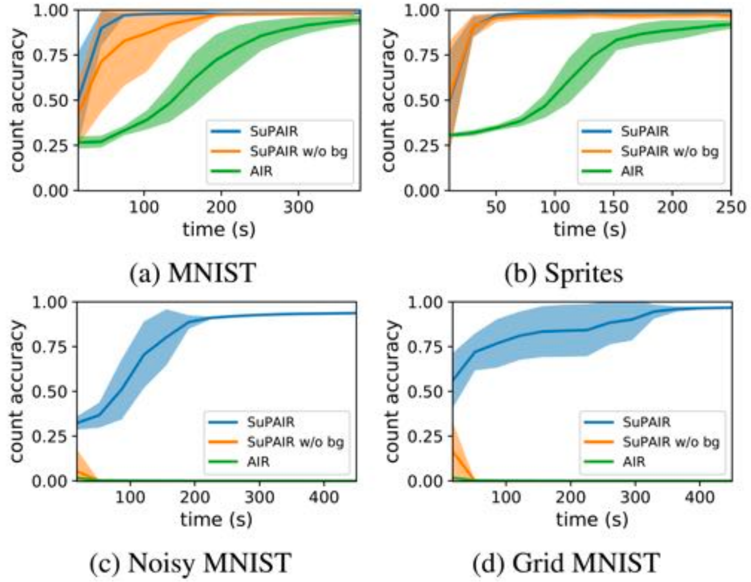
Thirty-sixth International Conference on Machine Learning

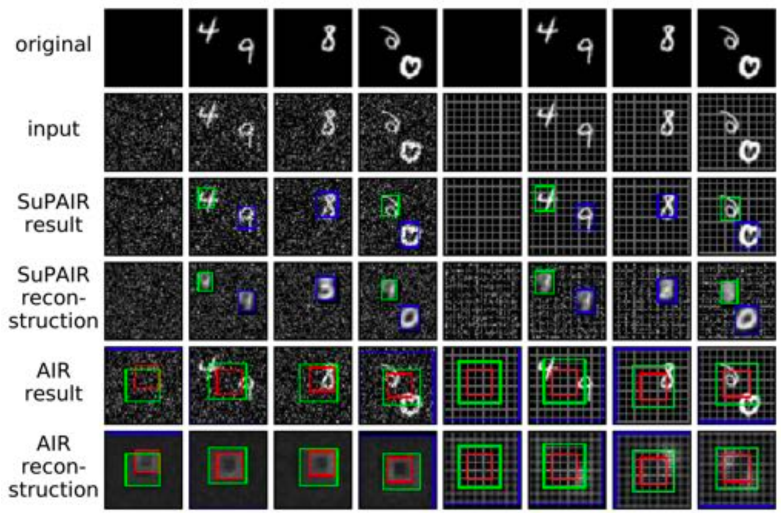
Consider e.g. unsupervised scene understanding using a generative model





[Attend-Infer-Repeat (AIR) model, Hinton et al. NIPS 2016]

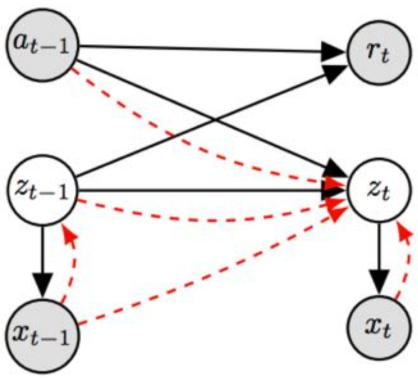


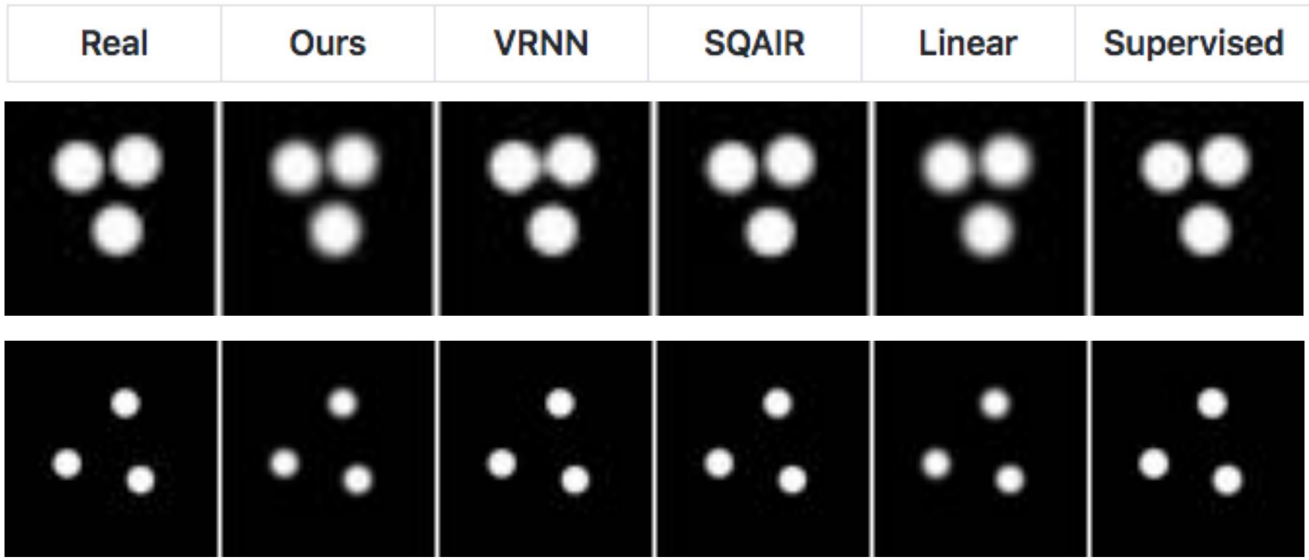


Replace VAE by SPN as object model

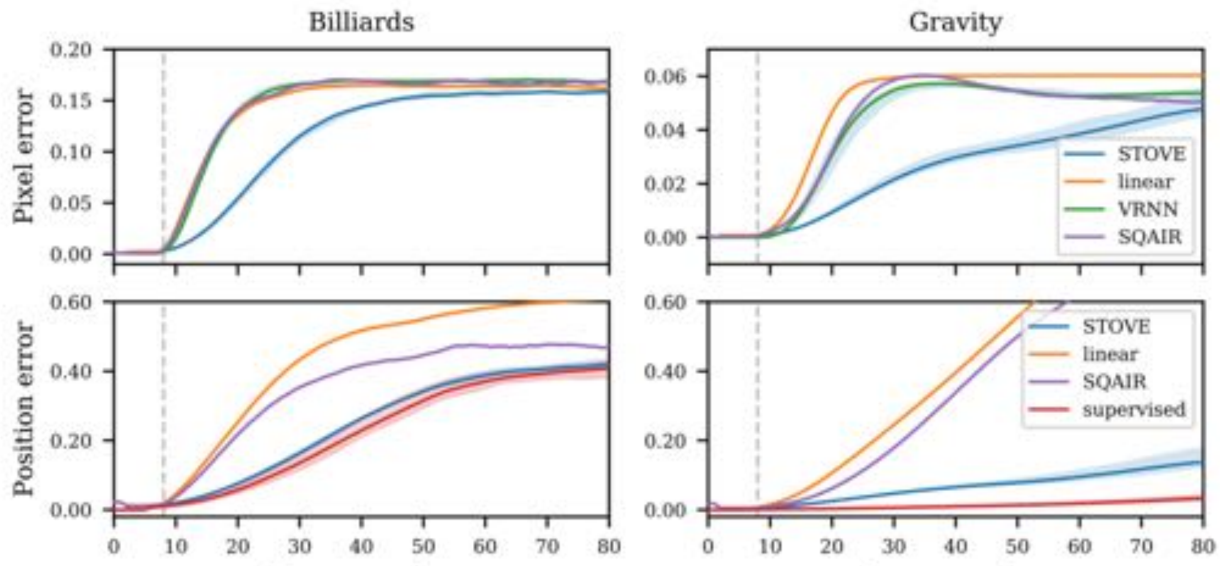
Unsupervised physics learning

[Kossen, Stelzner, Hussing, Voelcker, Kersting arXiv:1910.02425 2019]





putting structure and tractable inference into deep models



Programming languages for Systems AI,

the computational and mathematical modeling of complex AI systems.

[Laue et al. NeurIPS 2018; Kordjamshidi, Roth, Kersting: "Systems AI: A Declarative Learning Based Programming Perspective." IJCAI-ECAI 2018]

Th ... y

...bu ... e,
d

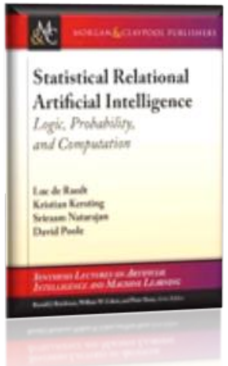
Eric Schmidt, Executive Chairman, Alphabet Inc.: Just Say "Yes", Stanford Graduate School of Business, May 2, 2017.https://www.youtube.com/watch?v=vbb-AjiXyh0.

Since science is more than a single table !

P(heart attack |  **)?**

Crossover of ML and AI with data & programming abstractions

De Raedt, Kersting, Natarajan, Poole: Statistical Relational Artificial Intelligence: Logic, Probability, and Computation. Morgan and Claypool Publishers, ISBN: 9781627058414, 2016.

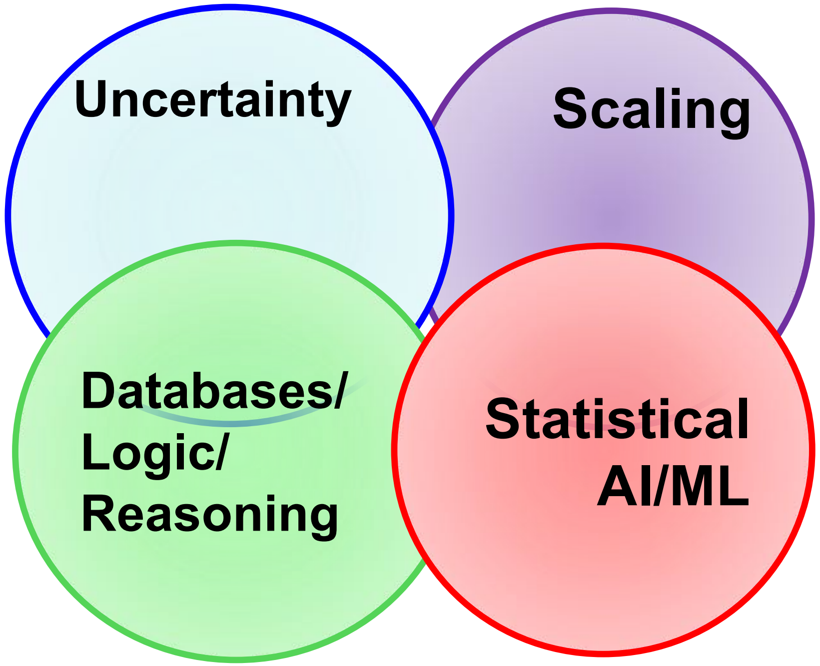


KATHOLIEKE UNIVERSITEIT LEUVEN
THE UNIVERSITY OF TEXAS AT DALLAS
TECHNISCHE UNIVERSITÄT DARMSTADT
UBC THE UNIVERSITY OF BRITISH COLUMBIA

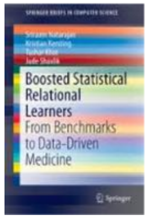
building general-purpose AI and ML machines

make the ML/AI expert more effective

increases the number of people who can successfully build ML/AI applications



Natarajan, Khot, Kersting, Shavlik. Boosted Statistical Relational Learners. Springer Brief 2015

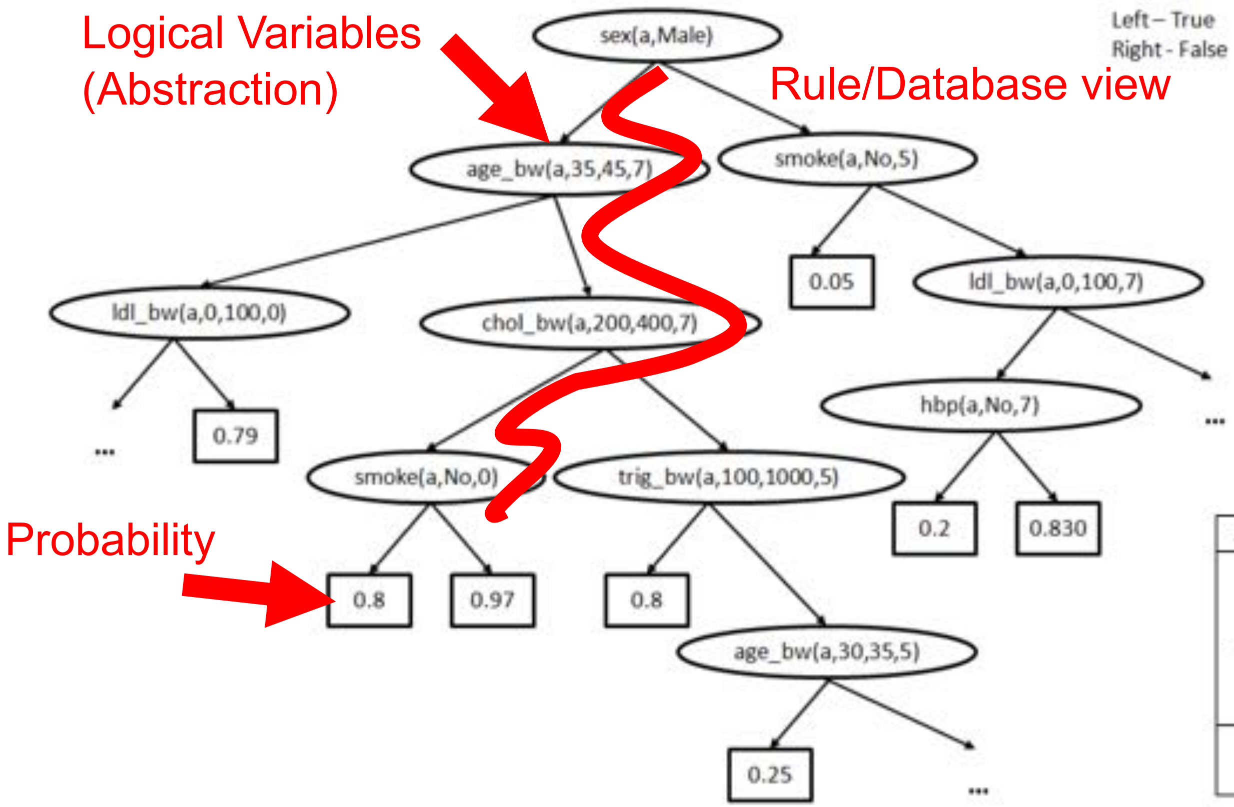


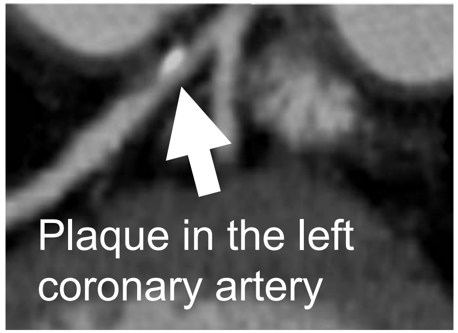
Understanding Electronic Health Records

Atherosclerosis is the cause of the majority of Acute Myocardial Infarctions (heart attacks)

TECHNISCHE UNIVERSITÄT DARMSTADT

UTD THE UNIVERSITY OF TEXAS AT DALLAS





[Circulation; 92(8), 2157-62, 1995; JACC; 43, 842-7, 2004]

Algorithm	Accuracy	AUC-ROC
J48	0.667	0.607
SVM	0.667	0.5
AdaBoost	0.667	0.608
Bagging	0.677	0.613
NB	0.75	0.653
RPT	0.669*	0.778
RFGB	0.667*	0.819

The higher, the better

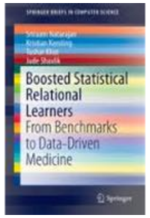
25%

Algorithm for Mining Markov Logic Networks	Likelihood The higher, the better	AUC-ROC The higher, the better	AUC-PR The higher, the better	Time The lower, the better
Boosting	0.81	0.96	0.93	**9s**
LSM	0.73	0.54	0.62	**93 hrs**

11% 78% 50% 37200x faster

state-of-the-art

[Kersting, Driessens ICML´08; Karwath, Kersting, Landwehr ICDM´08; Natarajan, Joshi, Tadepelli, Kersting, Shavlik. IJCAI´11; Natarajan, Kersting, Ip, Jacobs, Carr IAAI `13; Yang, Kersting, Terry, Carr, Natarajan AIME ´15; Khot, Natarajan, Kersting, Shavlik ICDM´13, MLJ´12, MLJ´15, Yang, Kersting, Natarajan BIBM`17]



Natarajan, Khot, Kersting, Shavlik. Boosted Statistical Relational Learners. Springer Brief 2015

TECHNISCHE UNIVERSITÄT DARMSTADT

UTD THE UNIVERSITY OF TEXAS AT DALLAS

https://starling.utdallas.edu/software/boostsrl/wiki/

StARLinG LAB — People | Publications | Projects | Software | Datasets | Blog

BOOSTSRL BASICS

- Getting Started
- File Structure
- Basic Parameters
- Advanced Parameters
- Basic Modes
- Advanced Modes

ADVANCED BOOSTSRL

- Default (RDN-Boost)
- MLN-Boost
- Regression
- One-Class Classification
- Cost-Sensitive SRL
- Learning with Advice
- Approximate Counting
- Discretization of Continuous-Valued Attributes
- Lifted Relational Random Walks
- Grounded Relational Random Walks

APPLICATIONS

- Natural Language Processing

BoostSRL Wiki

BoostSRL (Boosting for Statistical Relational Learning) is a gradient-boosting based approach to learning different types of SRL models. As with the standard gradient-boosting approach, our approach turns the model learning problem to learning a sequence of regression models. The key difference to the standard approaches is that we learn relational regression models i.e., regression models that operate on relational data. We assume the data in a predicate logic format and the output are essentially first-order regression trees where the inner nodes contain conjunctions of logical predicates. For more details on the models and the algorithm, we refer to our book on this topic.

Sriraam Natarajan, Tushar Khot, Kristian Kersting and Jude Shavlik, Boosted Statistical Relational Learners: From Benchmarks to Data-Driven Medicine . SpringerBriefs in Computer Science, ISBN: 978-3-319-13643-1, 2015

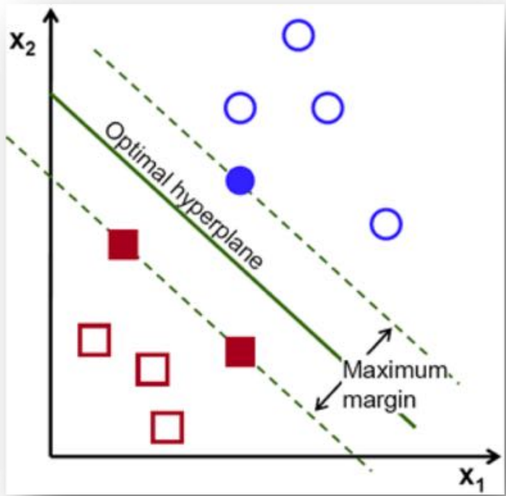
Human-in-the-loop learning

Not every scientist likes to turn math into code

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \mathcal{P}(\mathbf{w},b,\boldsymbol{\xi}) = \frac{1}{2}\mathbf{w}^2 + C\sum_{i=1}^{n}\xi_i$$

$$\text{subject to} \begin{cases} \forall i \quad y_i(\mathbf{w}^\top\Phi(\mathbf{x}_i)+b) \geq 1-\xi_i \\ \forall i \quad \xi_i \geq 0 \end{cases}$$

Support Vector Machines
Cortes, Vapnik MLJ 20(3):273-297, 1995



Kersting, Mladenov, Tokmakov AIJ´17, Mladenov, Heinrich, Kleinhans, Gonsio, Kersting DeLBP´16

High-level Languages for Mathematical Programs

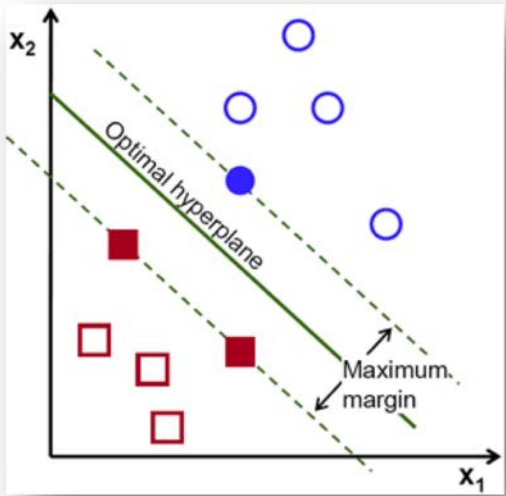
Write down SVM in „paper form.“ The machine compiles it into solver form.

```
#QUADRATIC OBJECTIVE
minimize: sum{J in feature(I,J)} weight(J)**2 + c1 * slack + c2 * coslack;

#labeled examples should be on the correct side
subject to forall {I in labeled(I)}: labeled(I)*predict(I) >= 1 - slack(I);

#slacks are positive
subject to forall {I in labeled(I)}: slack(I) >= 0;
```

Embedded within Python s.t. loops and rules can be used

reloop

RELOOP: A Toolkit for Relational Convex Optimization

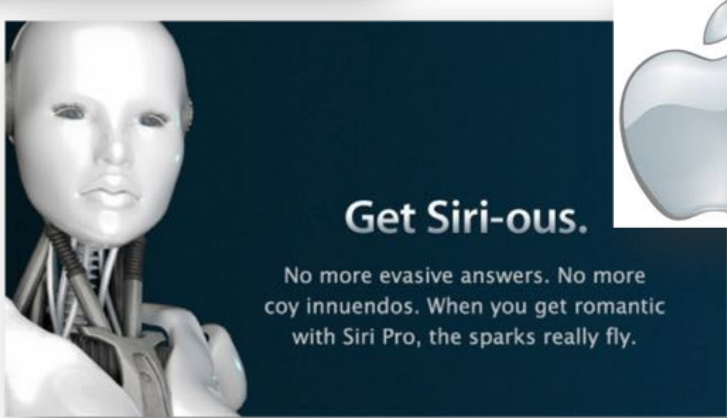


Support Vector Machines

Cortes, Vapnik MLJ 20(3):273-297, 1995

There are strong invests into high-level programming languages for AI/ML

RelationalAI, Apple, Microsoft and Uber are investing hundreds of millions of US dollars



UNIVERSITY OF CAMBRIDGE

UBER AI Labs

Max Planck Institute for Intelligent Systems

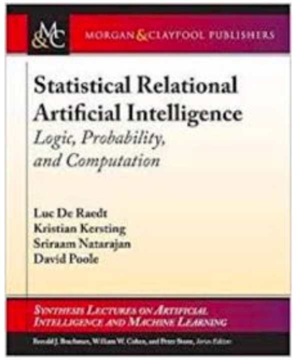
UNI GRAZ

FLOW

TECHNISCHE UNIVERSITÄT DARMSTADT

Microsoft Research

UBER AI Labs



relationalAI

AI for the enterprise

Getting deep systems that reason and know when they don’t know

Responsible AI systems that explain their decisions and co-evolve with the humans

Open AI systems that are easy to realize and understandable for the domain experts

„Tell the AI when it is right for the wrong reasons and it adapts ist behavior“

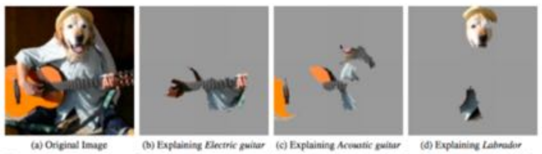


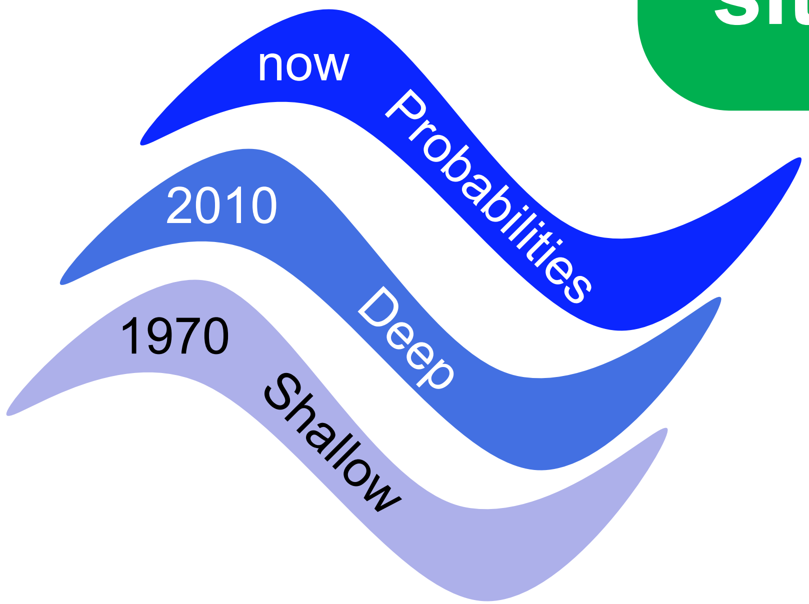
Figure 4: Explaining an image classification prediction made by Google’s Inception network, highlighting positive pixels. The top 3 classes predicted are “Electric Guitar” ($p = 0.32$), “Acoustic guitar” ($p = 0.24$) and “Labrador” ($p = 0.21$)

Teso, Kersting AIES 2019

AAAI / ACM conference on ARTIFICIAL INTELLIGENCE, ETHICS, AND SOCIETY

The third wave of differentiable programming

Getting deep systems that know when they do not know and, hence, recognise new situations and adapt to them



Overall, AI/ML/DS indeed refine "formal" science, but …

AI is more than deep neural networks. Probabilistic and causal models are whiteboxes that provide insights into applications

+ AI is more than a single table. Loops, graphs, different data types, relational DBs, … are central to ML/AI and high-level programming languages for ML/AI help to capture this complexity and makes using ML/AI simpler

+ AI is more than just Machine Learners and Statisticians: **AI is a team sport**

= The Third Wave of AI requires integrative CS, from software engineering and DB systems, over ML and AI to computational CogSci

A lot left to be done

But AI and Humans can and will be partners!