Deep Learning 2.0: AI that Builds AI

Frank Hutter

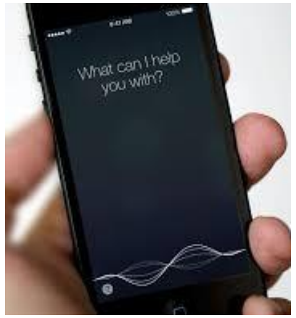
Professor for Machine Learning
University of Freiburg
fh@cs.uni-freiburg.de

@FrankRHutter

These slides are available at www.automl.org/talks

Motivation: Deep Learning is Everywhere Now

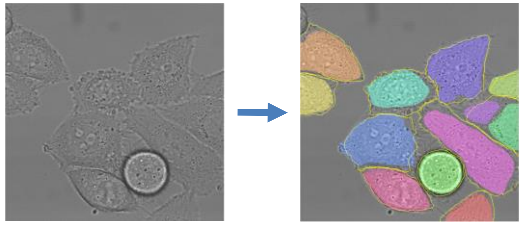
Speech recognition

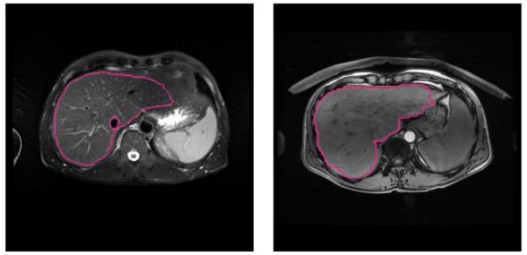


Auto-Translator

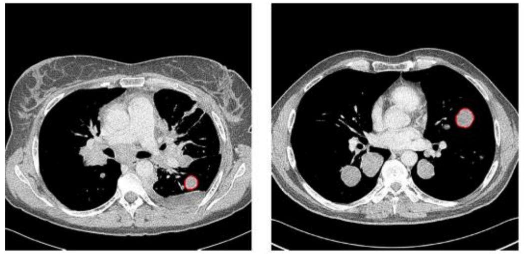
Recommender systems

Semantic segmentation





Autonomous Driving



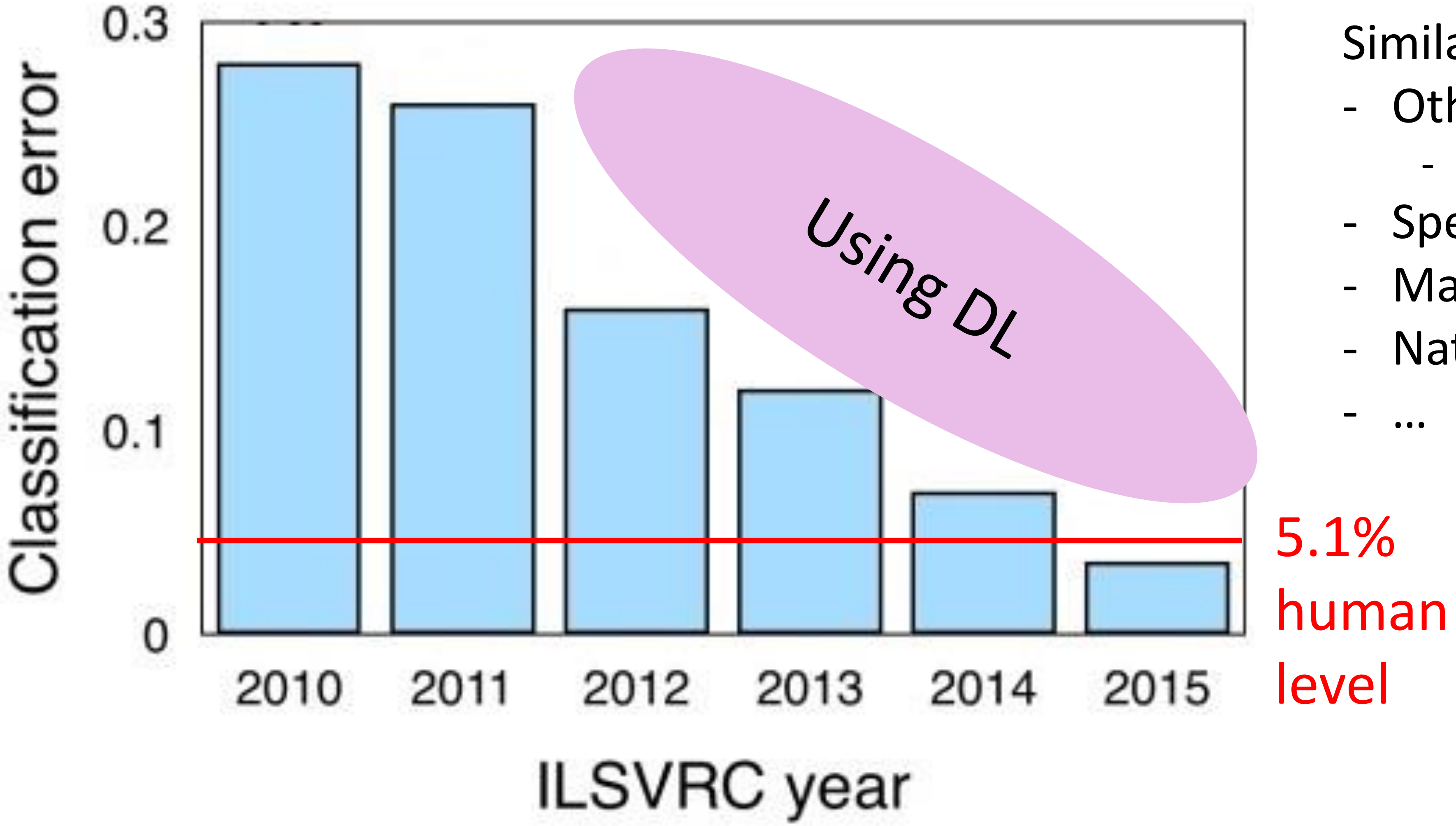
Applications in medicine

AI-generated art

OpenAI's DALL-E 2: "A painting of an orthopedist diagnosing a patient's knee injury"

Game playing

Massive Progress in Object Recognition in the Last Decade



Similar progress in:
- Other computer vision tasks
  - E.g., semantic segmentation
- Speech recognition
- Machine translation
- Natural language processing
- …

Goals of Today's Lecture

- Basic Principles of "Traditional" Machine Learning
  - Supervised Learning (Classification, Regression)
  - Machine Learning Design Cycle
  - Proper Evaluation Protocols
- Traditional Machine Learning vs. Deep Learning
  - Learning Features From Raw Data
- From Deep Learning to Deep Learning 2.0
  - AutoML: Efficient ML & DL at the Push of a Button
  - Learning Entire Algorithms: AI that Builds AI

Supervised Learning: The Basic Idea

- Use past experience to predict the future
  - Use labelled data points $\langle(\mathbf{x}_i, y_i)\rangle_{i=1}^N$ that we collected in the past
  - to automatically construct a model whose prediction $\hat{y}_{N+1}$ for a new data point $\mathbf{x}_{N+1}$ is close to the actual label y_{N+1}.

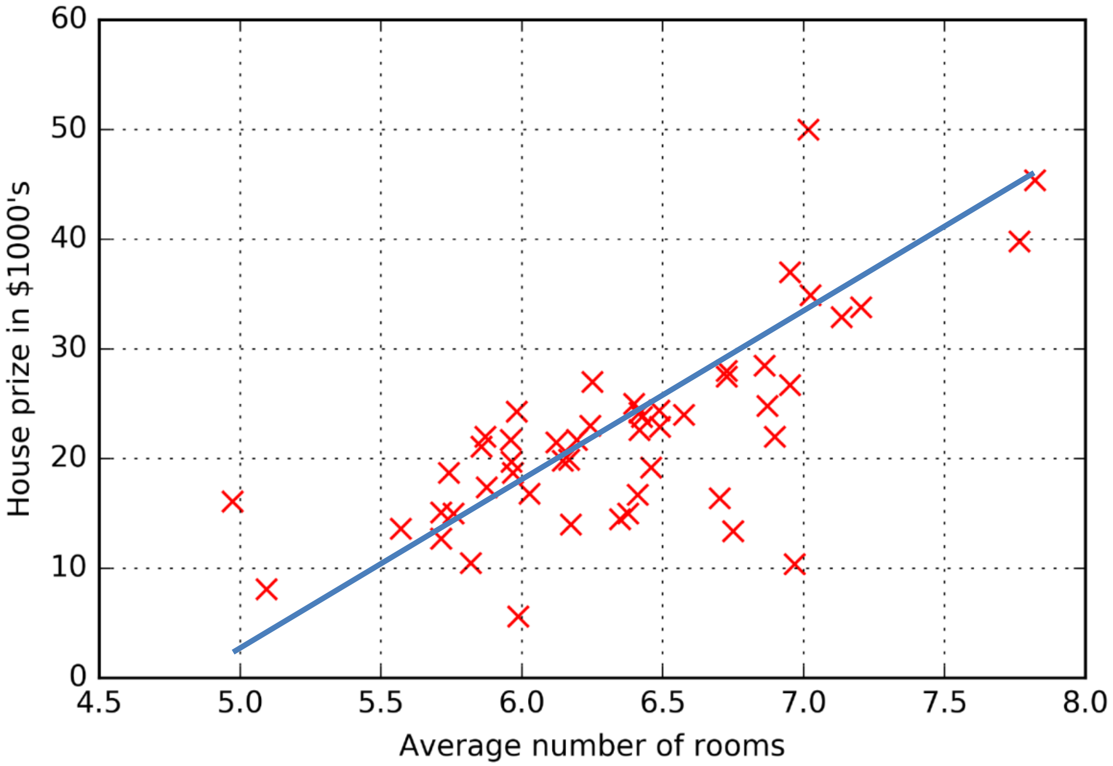
- Machine learning terminology:
  - Data point $\mathbf{x}_i$, often a vector in $\mathbb{R}^D$
  - Label y_i
    - Regression: $y_i \in \mathbb{R}$
    - Classification: y_i discrete, e.g. $y_i \in \{$true, false$\}$, or $y_i \in \{$German, English, Spanish$\}$
  - Past experience = training set $\langle(\mathbf{x}_i, y_i)\rangle_{i=1}^N$
  - Automatically construct = learn, fit, induce
  - Model = function, hypothesis, classifier/regressor

Supervised Learning: A Simple Regression Example

Predicting housing prices

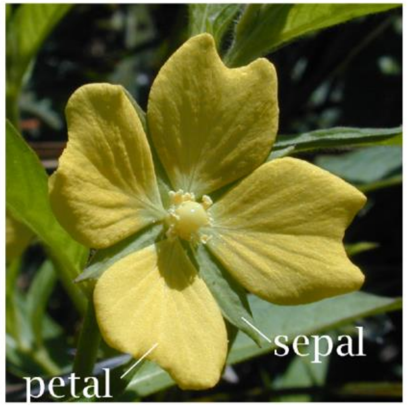
- Let's say we only know the average number of rooms in an area
- And we'd like to predict the prize for a house in that area
- One data point: number of rooms $\mathbf{x}_i$ and its prize y_i (in 1000's)
- This is a regression problem since $y_i \in \mathbb{R}$

avg. # rooms	y_i
6.575	24
6.377	21.6
5.57	34.7
5.713	33.4
7.024	36.2
5.963	28.7
5.741	22.9
6.417	27.1
6.727	16.5
6.417	18.9
...	...



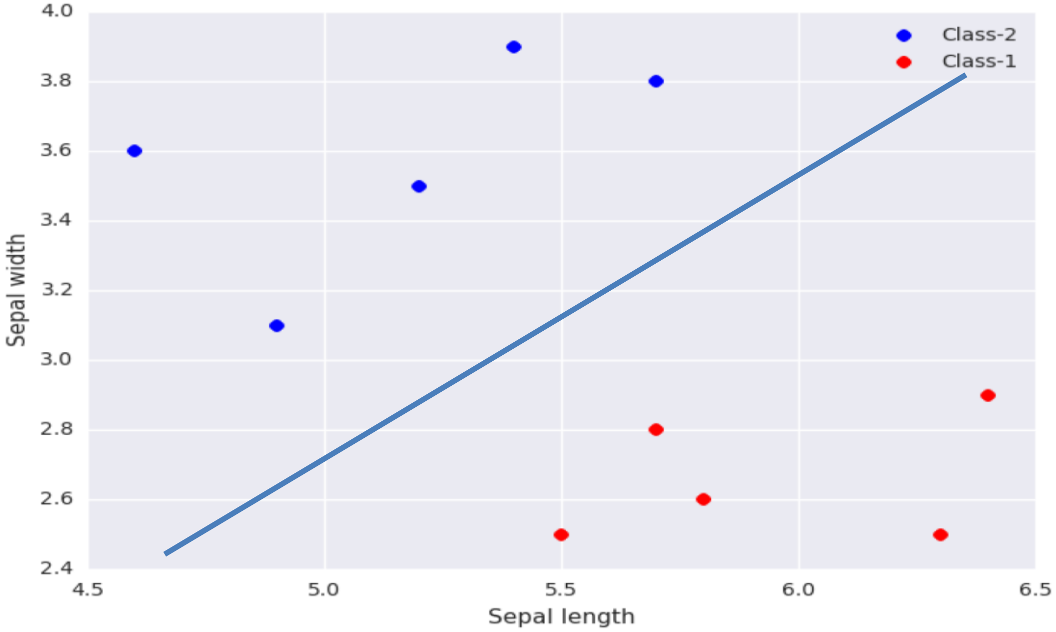
Supervised Learning: a Simple Classification Example

- A classical data set from Botany: classifying Iris flowers
  - feature 1: sepal length
  - feature 2: sepal width



Classification problem:
determine the flower's type (out of 3 options)

$\mathbf{x}_{i,1}$	$\mathbf{x}_{i,2}$	y_i
6.40	2.90	2
5.50	2.50	2
5.20	3.50	1
4.60	3.60	1
5.70	3.80	1
6.30	2.50	2
5.80	2.60	2
4.90	3.10	1
5.70	2.80	2
5.40	3.90	1



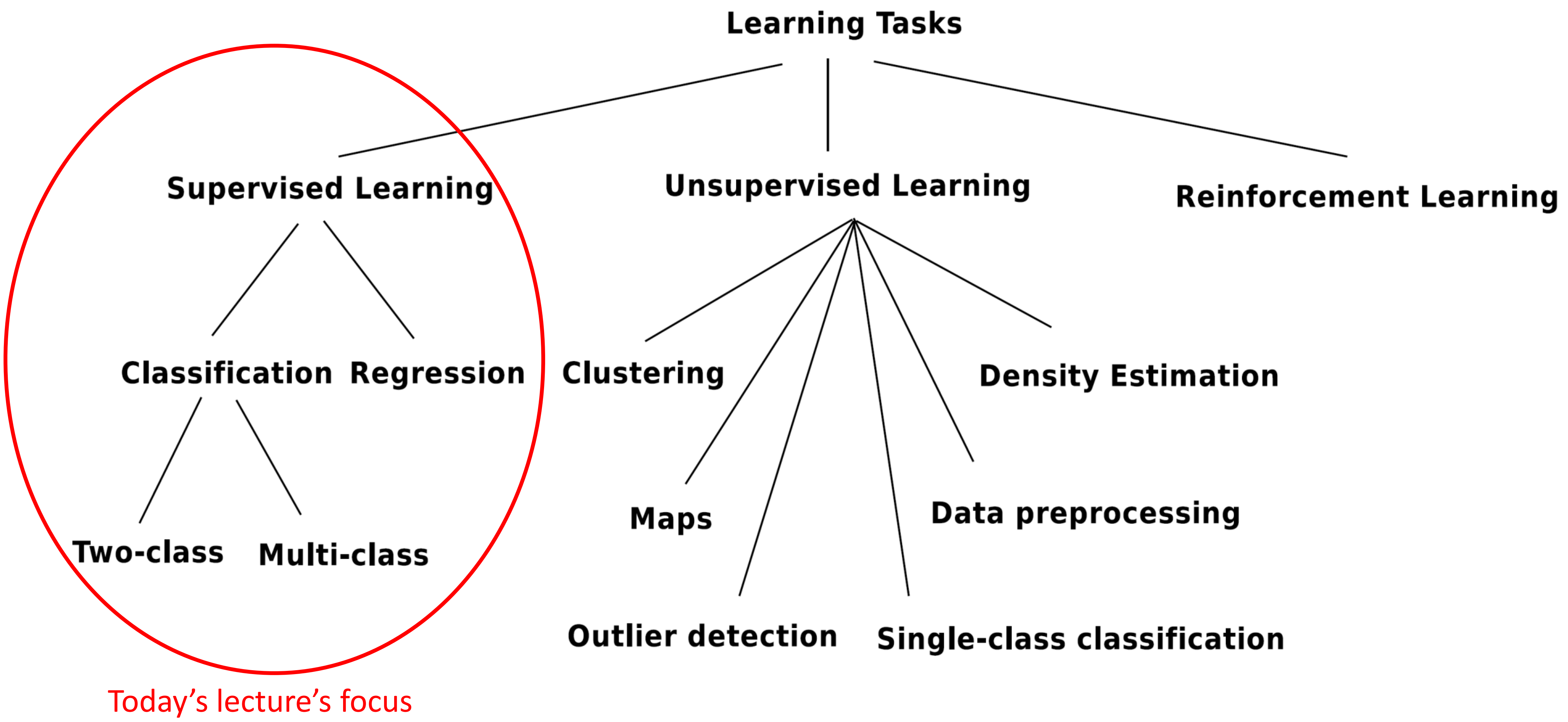
An Interactive Example

- A private customer comes to a bank and applies for a loan
  - Should the bank give him the loan?
  - What features should the bank use to decide?
    - Features: credit amount, income, age, etc.

- Ways of casting this as a machine learning problem:
  - 1. Should the bank give the loan?
    - Classification or regression?

  - 2. How much interest should the bank charge?
    - Classification or regression?

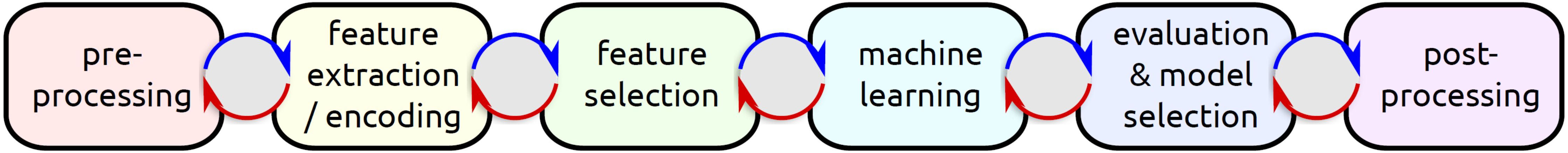
Types of Learning Tasks



Outline

- Basic Principles of “Traditional” Machine Learning
  - Supervised Learning (Classification, Regression)
  - Machine Learning Design Cycle
  - Proper Evaluation Protocols
- Traditional Machine Learning vs. Deep Learning
  - Learning Features From Raw Data
- From Deep Learning to Deep Learning 2.0
  - AutoML: Efficient ML & DL at the Push of a Button
  - Learning Entire Algorithms: AI that Builds AI

The Machine Learning Design Cycle



- It's all about the data
  - Domain experts needed: curation, preprocessing, feature extraction & selection
- Machine learning
  - Focus for us; ML methods development
- Evaluation & model selection
  - Focus for us; I'm developing automated methods to do this better
- Post-processing
  - Domain experts needed: is the model actually useful in practice? Iterate!

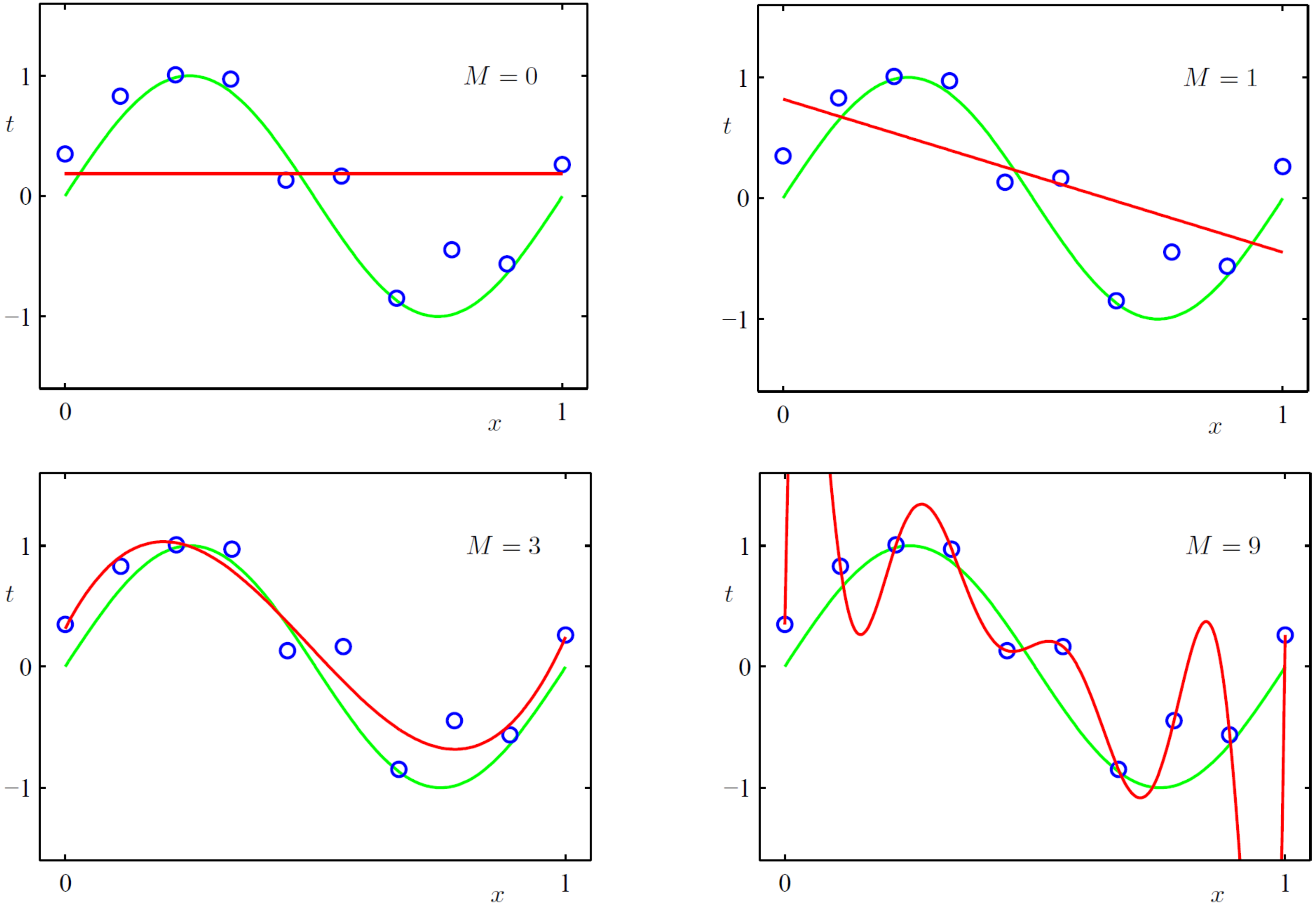
Outline

- Basic Principles of "Traditional" Machine Learning
  - Supervised Learning (Classification, Regression)
  - Machine Learning Design Cycle
  - Proper Evaluation Protocols
- Traditional Machine Learning vs. Deep Learning
  - Learning Features From Raw Data
- From Deep Learning to Deep Learning 2.0
  - AutoML: Efficient ML & DL at the Push of a Button
  - Learning Entire Algorithms: AI that Builds AI

Beware of Overfitting: What is the Best Explanation for the Data?

- What is the best fit for the data?

- M=9 has zero training error
  - But it does NOT generalize



The Difference Between Training and Test Set Performance

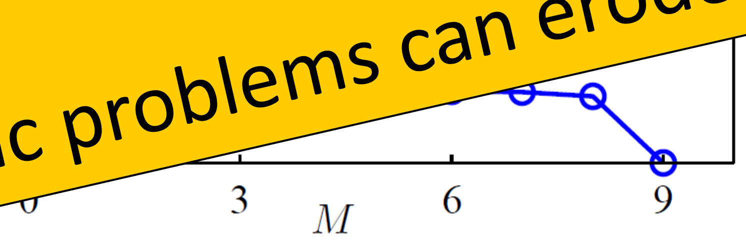
We care about good generalization, i.e. accurate predictions for new data

- Quantify dependence of generalization perform
  test set

A very common beginner's error:
reporting training errors instead of test errors

Those error rates cannot be achieved for new test data

Such basic problems can erode trust in AI



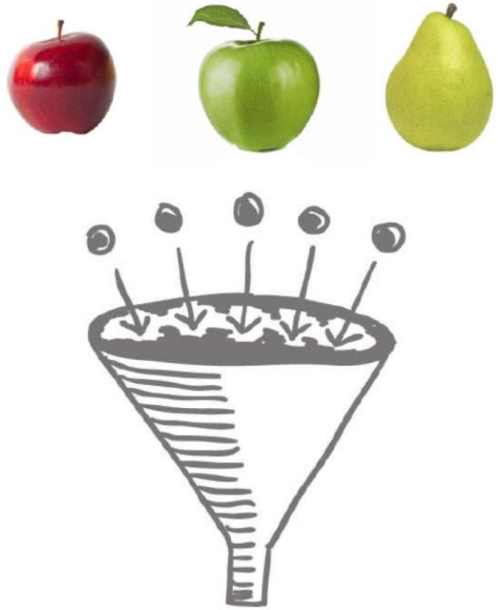
- Overfitting: *Fitting the data more than is warranted*

Outline

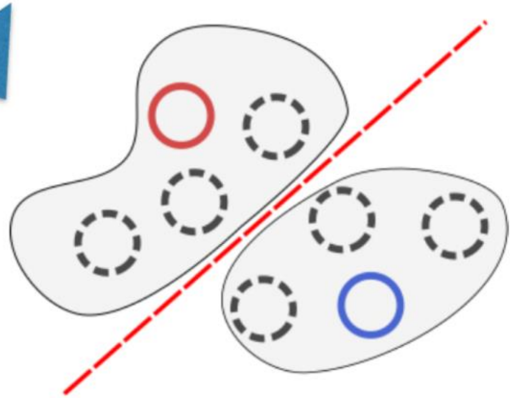
- Basic Principles of “Traditional” Machine Learning
  - Supervised Learning (Classification, Regression)
  - Machine Learning Design Cycle
  - Proper Evaluation Protocols
- Traditional Machine Learning vs. Deep Learning
  - Learning Features From Raw Data
- From Deep Learning to Deep Learning 2.0
  - AutoML: Efficient ML & DL at the Push of a Button
  - Learning Entire Algorithms: AI that Builds AI

Standard Machine Learning

- Standard machine learning algorithms are based on features
  - These are high-level attributes defined by domain experts
  - This requires (often substantial) feature engineering

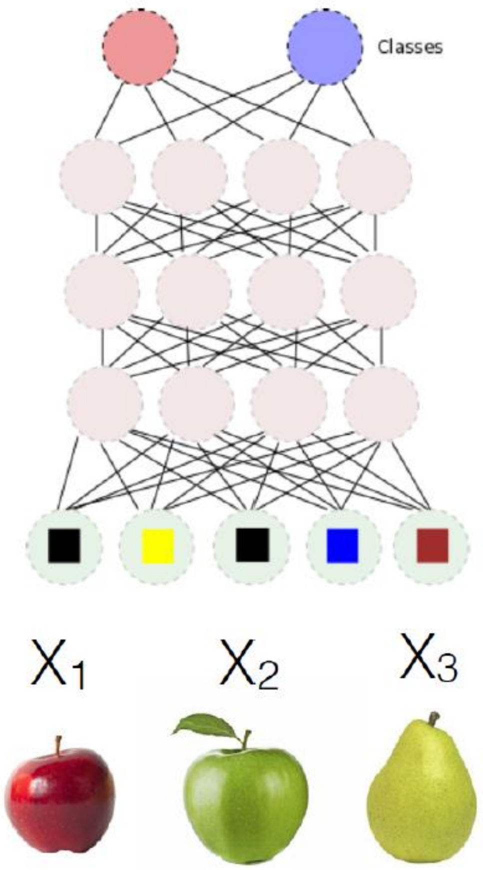


Features		Label
red	3.5 cm	apple
green	4 cm	apple
yellow	5cm	pear

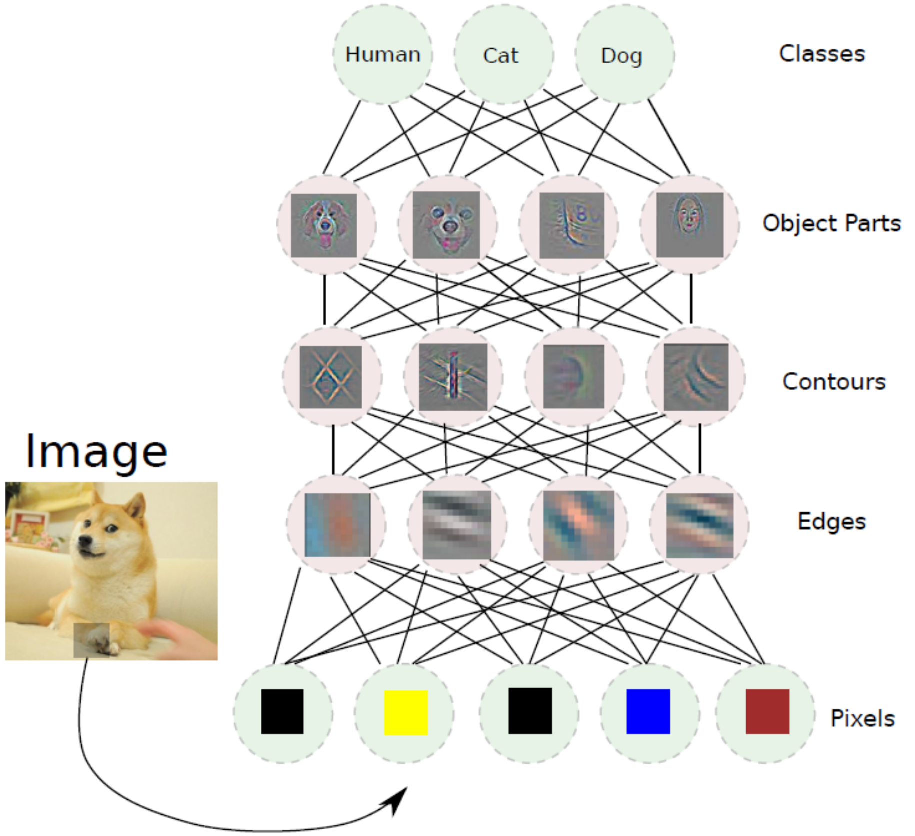


Deep Learning

- Jointly learn features and classifier, directly from raw data
- This is also referrred to as end-to-end learning

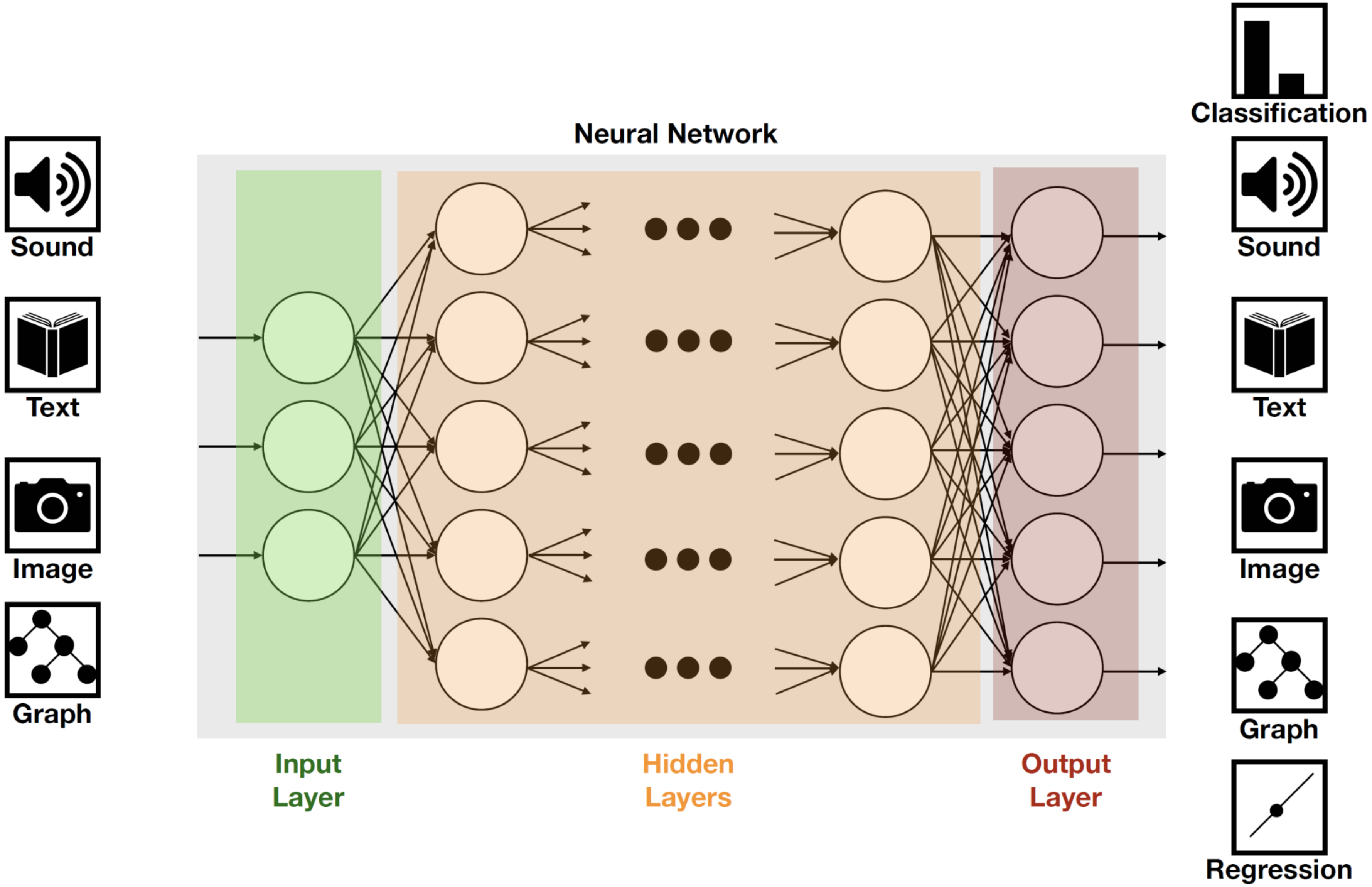


Deep Learning



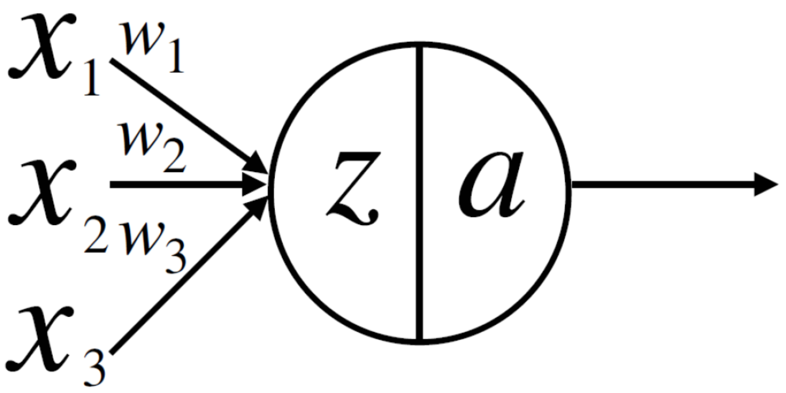
- Deep Learning: learning a hierarchy of representations that build on each other, from simple to complex
- Features are learned in an end-to-end fashion, from raw data

Basic Structure of a Neural Network



What Happens Under the Hood?

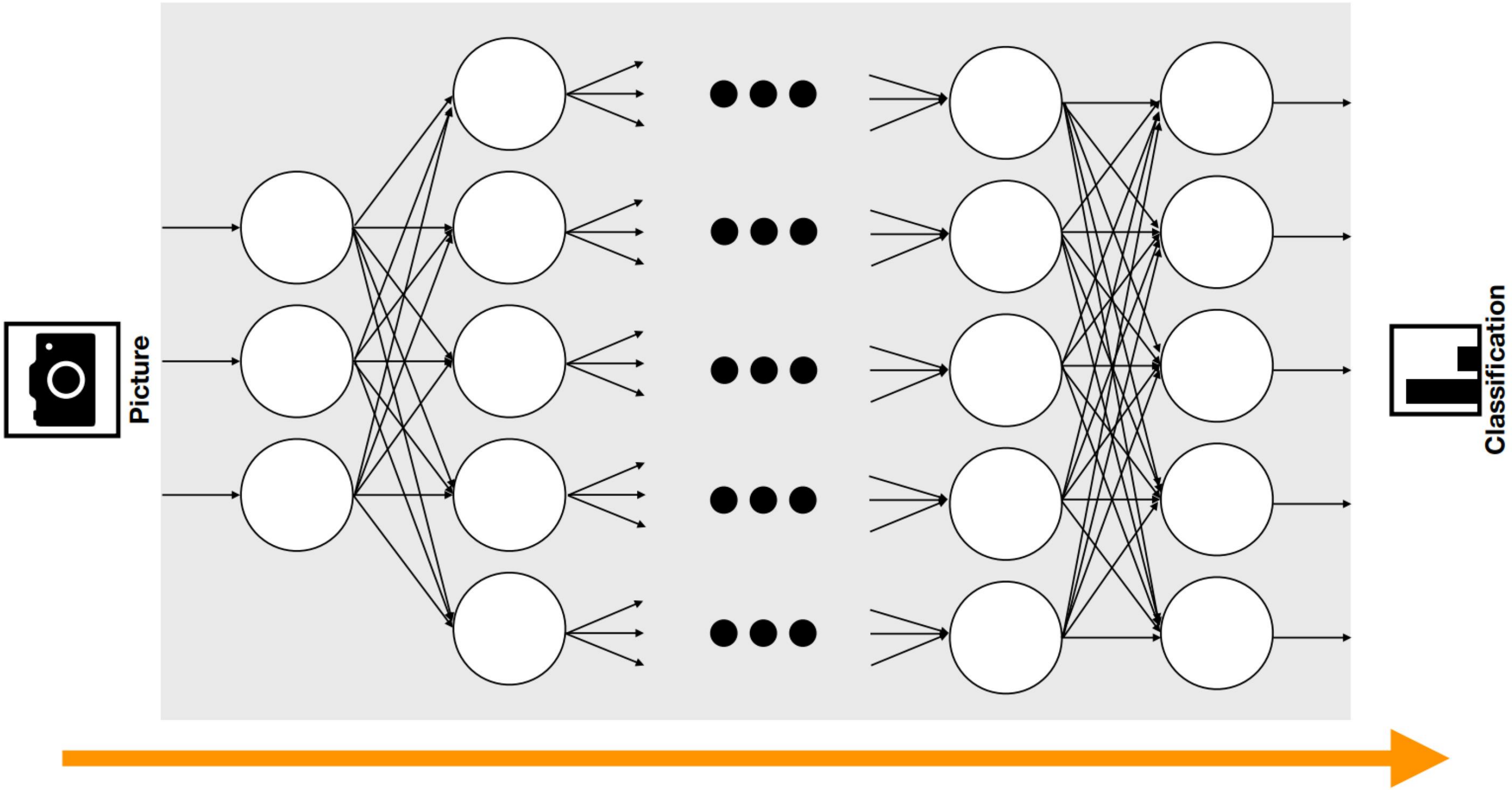
- A single neuron performs two simple steps of computation:



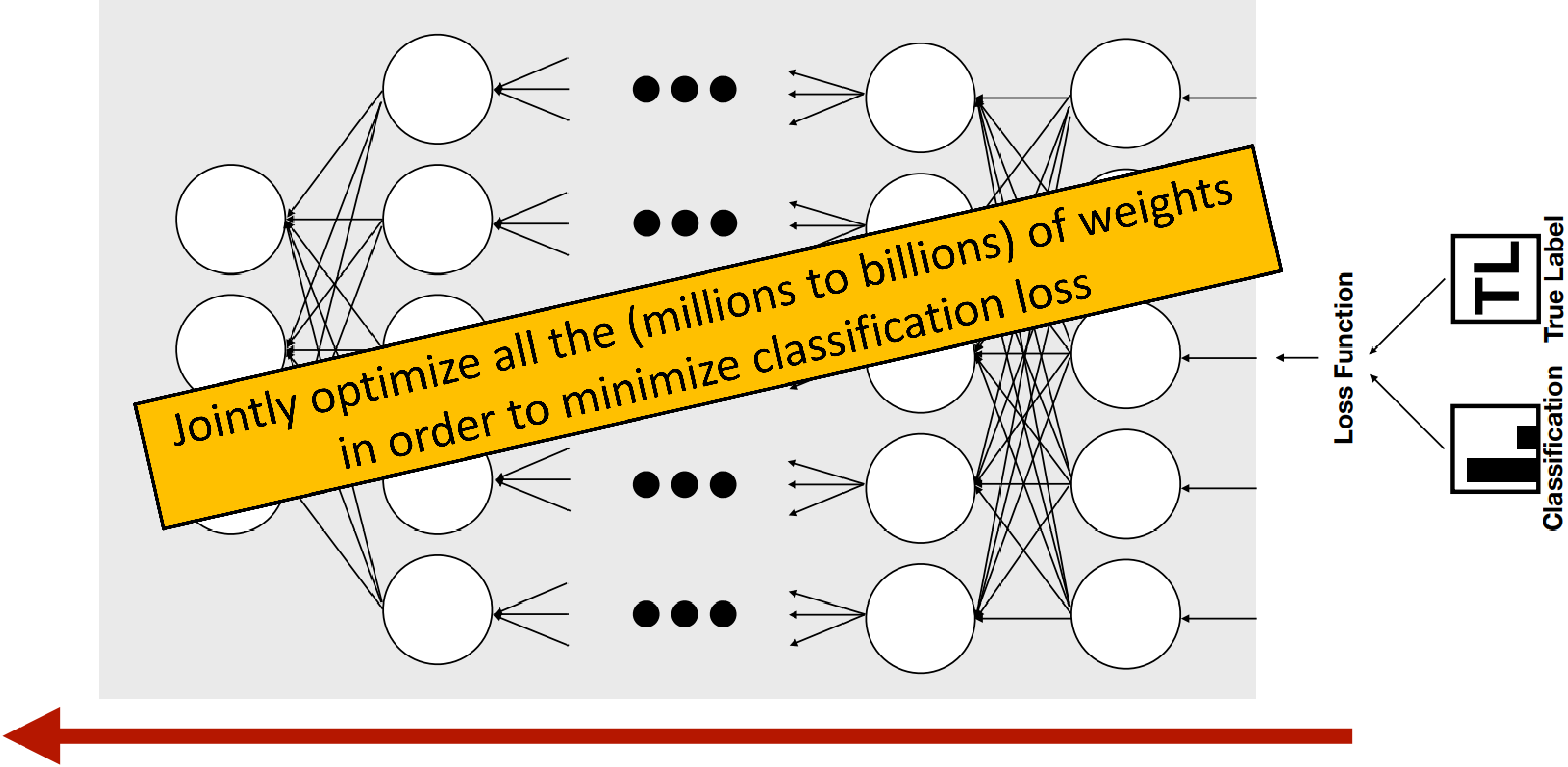
1. Compute a weighted sum of the inputs: $z = x_1 w_1 + x_2 w_2 + x_3 w_3$

2. Perform a nonlinear transformation: $a = h(z)$.

Information Flow Through a Neural Network – Forward Pass



Information Flow Through a Neural Network – Backward Pass



Why Deep Learning succeeded

Traditional ML practice before Deep Learning

Data → Domain expert engineers features → Traditional ML (XGBoost etc)

trial & error

Deep Learning

Data → Deep Learning (end-to-end)

- saves manual labour
- better results by end-to-end joint optimization

Outline

- Basic Principles of "Traditional" Machine Learning
  - Supervised Learning (Classification, Regression)
  - Machine Learning Design Cycle
  - Proper Evaluation Protocols
- Traditional Machine Learning vs. Deep Learning
  - Learning Features From Raw Data
- From Deep Learning to Deep Learning 2.0
  - AutoML: Efficient ML & DL at the Push of a Button
  - Learning Entire Algorithms: AI that Builds AI

erc European Research Council

2016 ERC Starting Grant AutoML

2021 ERC Consolidator Grant Deep Learning 2.0

From Deep Learning 1.0 to Deep Learning 2.0

Traditional ML practice before Deep Learning

Data

→ Domain expert engineers features → Traditional ML (XGBoost etc)

trial & error

Deep Learning

Data

→ DL expert chooses architecture & hyperparameters → Deep Learning (end-to-end)

trial & error

From Deep Learning 1.0 to Deep Learning 2.0

Deep Learning 2.0

Data

✅ saves human labour

✅ better results by
end-to-end joint optimization

→ Deep Learning 2.0 (end-to-end)

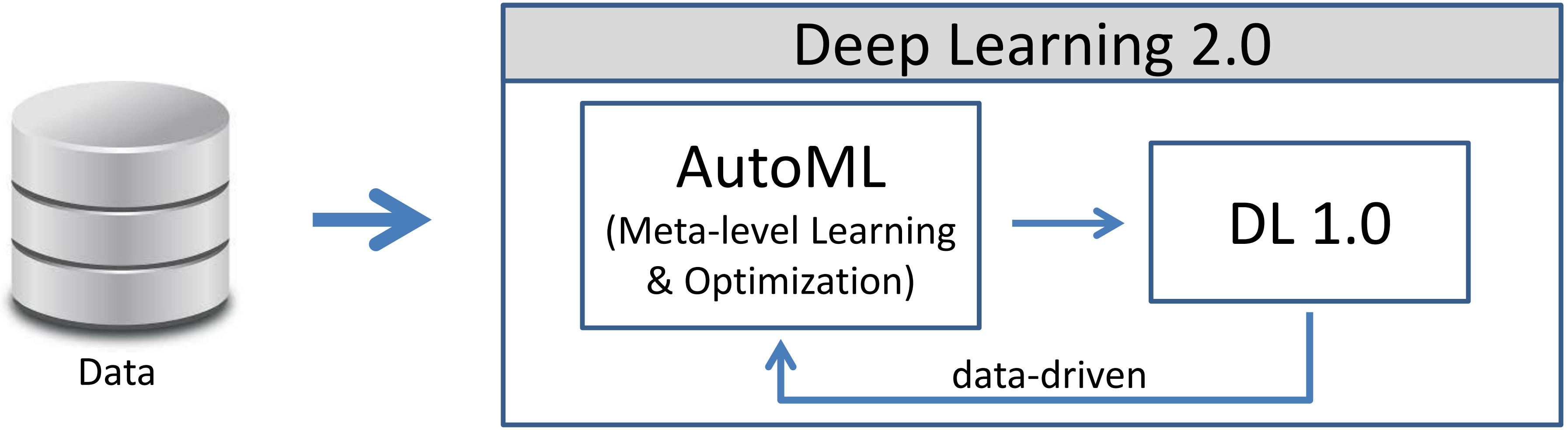
Deep Learning 1.0

Data

→ DL expert chooses architecture & hyperparameters → Deep Learning (end-to-end)

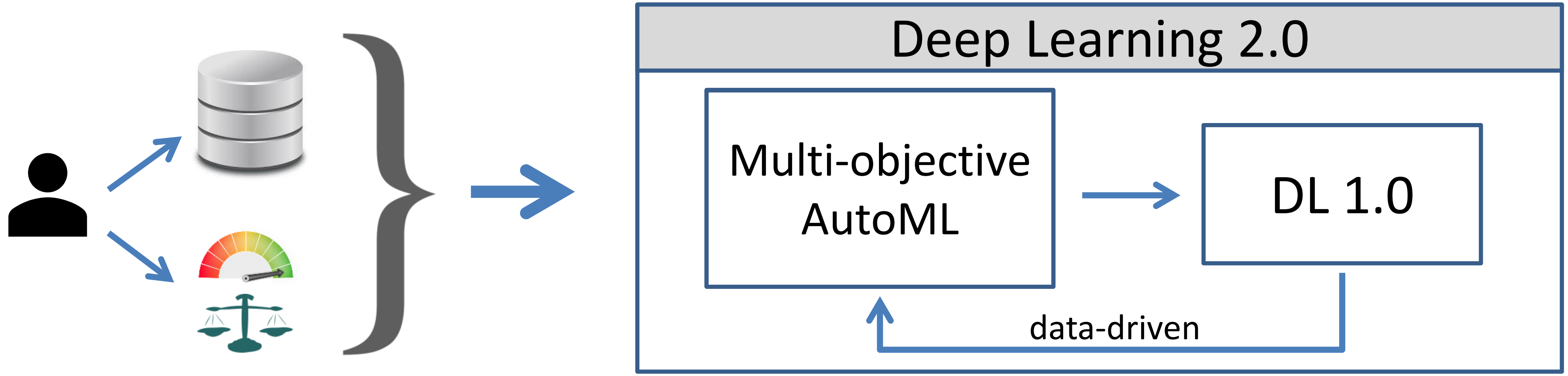
trial & error

From Deep Learning 1.0 to Deep Learning 2.0



- ✖ fairness
- ✖ robustness
- ✖ model calibration
- ✖ interpretability
- ✖ latency of predictions
- ✖ size(memory) of the model

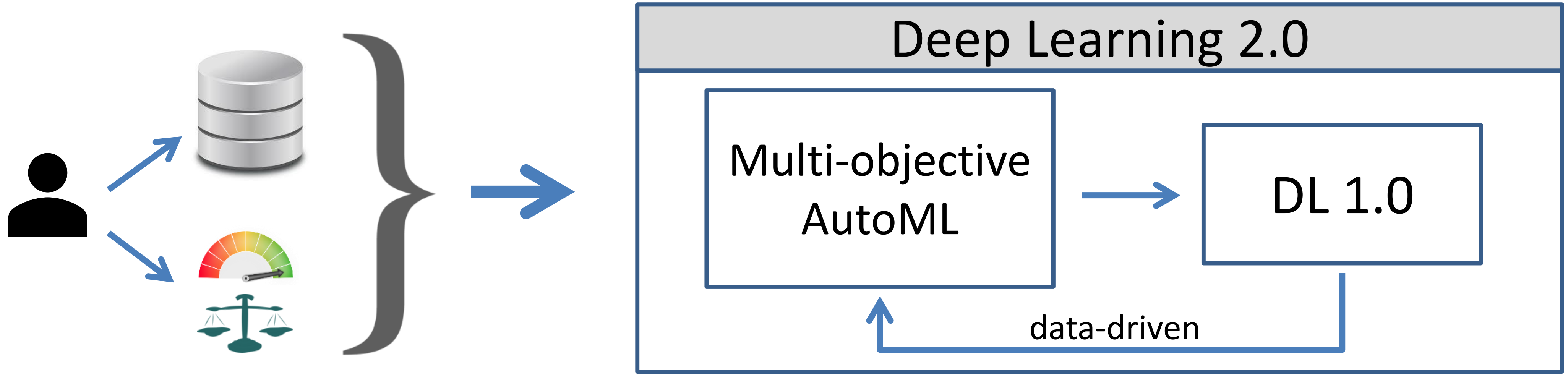
From Deep Learning 1.0 to Deep Learning 2.0



- domain expert can specify objectives

- fairness
- robustness
- model calibration
- interpretability
- latency of predictions
- size(memory) of the model

Expected Impact of Deep Learning 2.0



- **Paradigm-changing: democratizing Deep Learning**
  - DL 2.0 projects possible without a DL expert
  - DL 2.0 directly optimizes for user‘s objectives
    → Trustworthy AI by design

DL 2.0 will be even more pervasive than DL 1.0, with huge impact on the billion-dollar DL market

Outline

- Basic Principles of "Traditional" Machine Learning
  - Supervised Learning (Classification, Regression)
  - Machine Learning Design Cycle
  - Proper Evaluation Protocols
- Traditional Machine Learning vs. Deep Learning
  - Learning Features From Raw Data
- From Deep Learning to Deep Learning 2.0
  - AutoML: Efficient ML & DL at the Push of a Button
  - Learning Entire Algorithms: AI that Builds AI

European Research Council

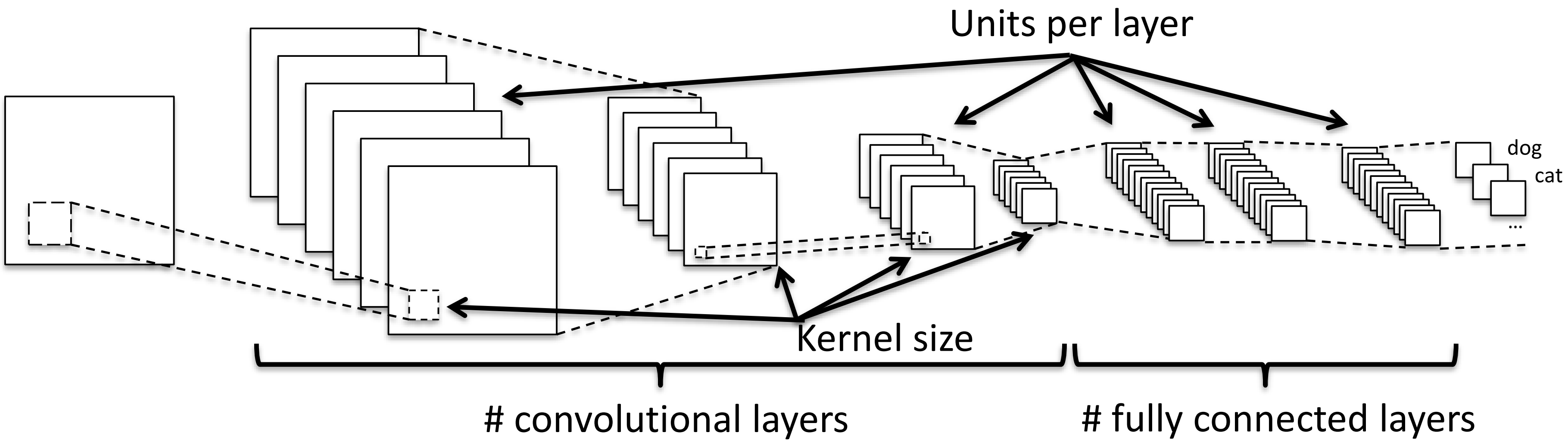
2016 ERC Starting Grant AutoML

2021 ERC Consolidator Grant Deep Learning 2.0

Choices in Deep Learning That AutoML Can Help With

Performance is very sensitive to many hyperparameters

– Architectural hyperparameters

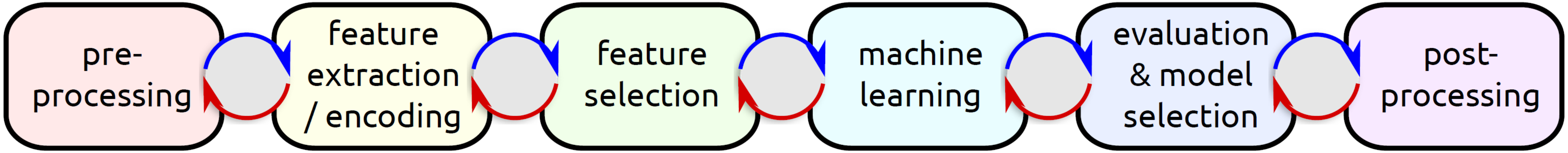


– Optimization: SGD variant, learning rate schedule, momentum, batch sizes, …

– Regularization: dropout rates, weight decay, data augmentation, …

→ Easily 20-50 design decisions

Choices in Traditional ML That AutoML Can Help With



- Clean & preprocess the data
- Select / engineer better features
- Select a model family
- Set the hyperparameters
- Construct ensembles of models
- …

Different Types of AutoML

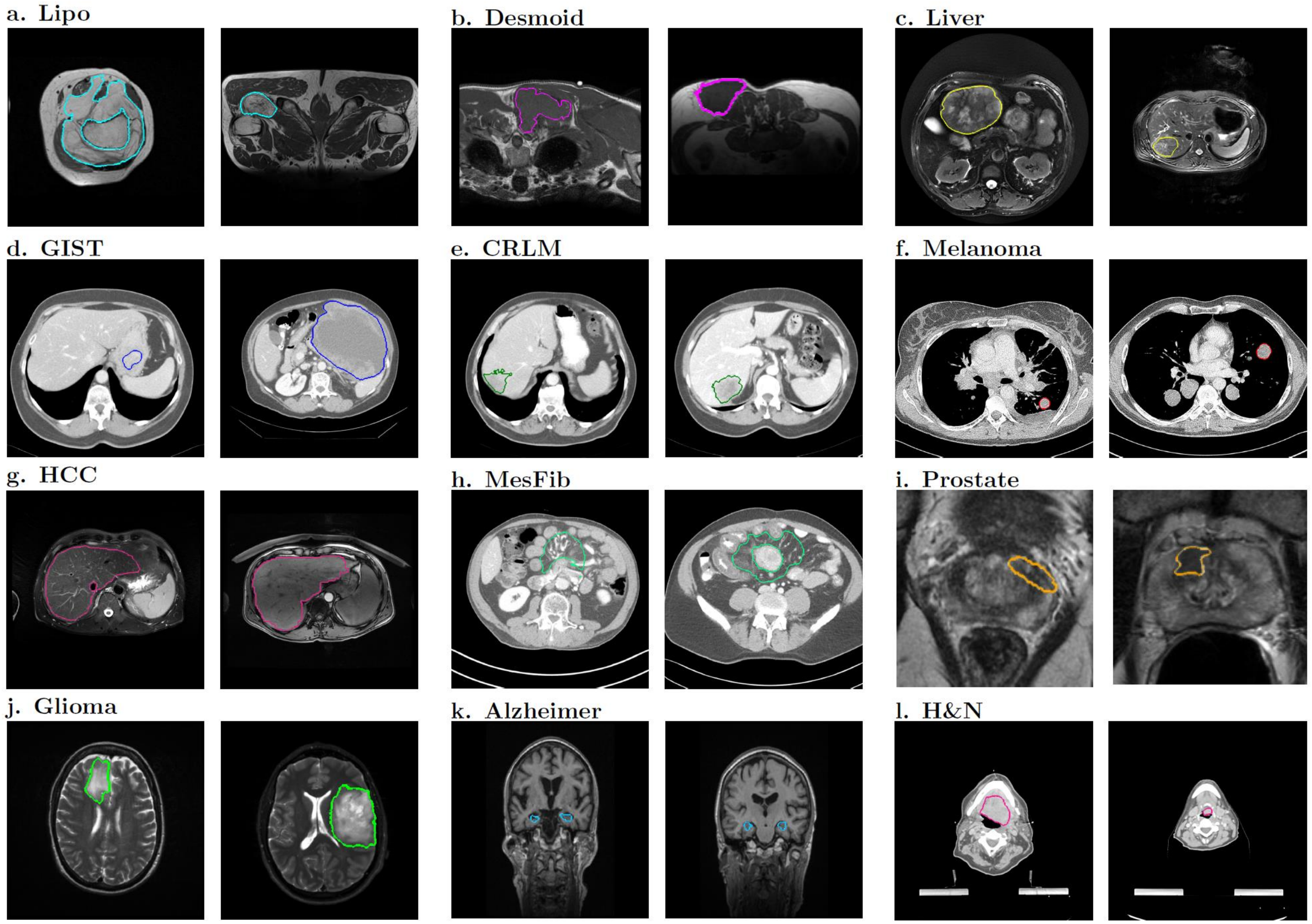
- Full AutoML Systems for Featurized Data
  - You have featurized data and just need a model
  - We’re world-leading in this, having won two world championship titles
    - E.g., better than 130 teams of human experts
  - Auto-sklearn: over 1000 forks on Github, 6000 stars, 20000 monthy downloads

Tool: Auto-sklearn, available at
https://github.com/automl/auto-sklearn

- Hyperparameter Optimization
  - You have data and a good pipeline for the data
  - But there are free choices you still need to set

Tool: SMAC, available at
https://github.com/automl/SMAC3

Case Study: AutoML Improved 12 Radiomics Segmentation Datasets



[Starmans et al, 2022]

(Department of Radiology and Nuclear Medicine, Erasmus MC, Rotterdam, the Netherlands)

Outline

- Basic Principles of “Traditional” Machine Learning
  – Supervised Learning (Classification, Regression)
  – Machine Learning Design Cycle
  – Proper Evaluation Protocols
- Traditional Machine Learning vs. Deep Learning
  – Learning Features From Raw Data
- From Deep Learning to Deep Learning 2.0
  – AutoML: Efficient ML & DL at the Push of a Button
  – Learning Entire Algorithms: AI that Builds AI

European Research Council

2016 ERC Starting Grant AutoML
2021 ERC Consolidator Grant Deep Learning 2.0

Deep Learning for Small Tabular Datasets

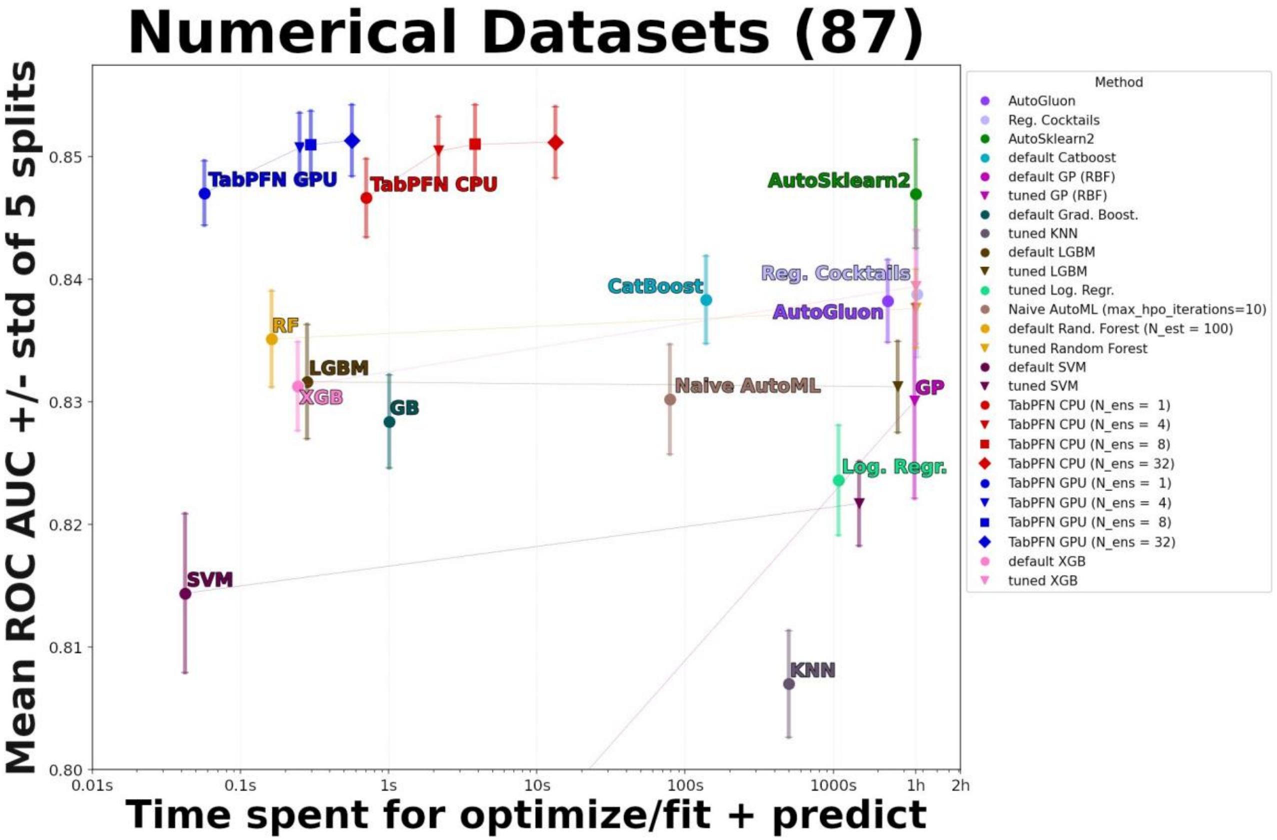
- Deep Learning Used to Have Many Problems for Tabular Datasets
  - Overfitting
  - Long training times

- Traditional ML Techniques Used to Dominate
  - Support Vector Machines
  - Decision Trees
  - Random Forests
  - Gradient Boosting (XGBoost)
  - Our Auto-sklearn

TabPFN: a Learned Algorithm For Small Tabular Data

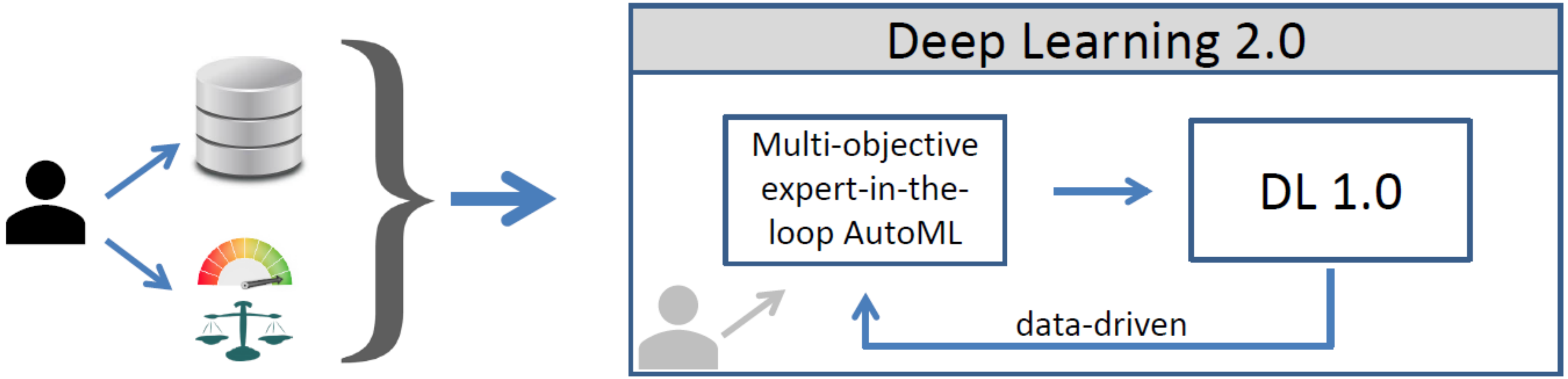
- TabPFN is a new state-of-the-art algorithm for tabular data
  - It is encoded in the weights of a neural network
  - It makes the best predictions for small numerical tabular data in 1 second

- Current limitations
  - Size: up to 1000 data points, 100 features, 10 classes
  - Not (yet) designed for: categorical features, missing values, uninformative features
  - High inference time



Take-aways

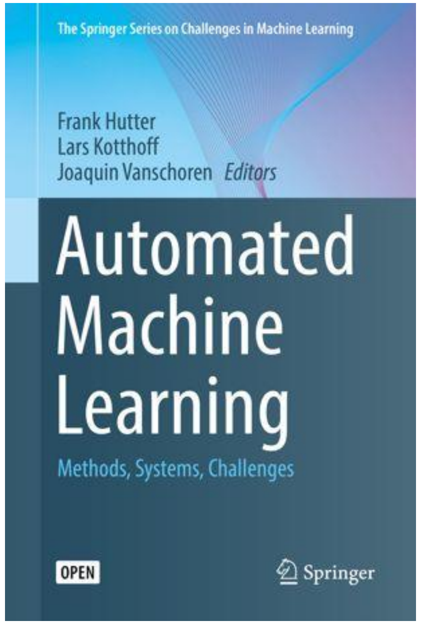
Deep Learning 2.0: expert-guided Auto-DL for the objectives at hand



1. Basics of ML
2. Basics of DL
3. Deep Learning 2.0
   - DL 2.0 projects possible without a DL expert
   - Strong open-source tools are already available
   - DL 2.0 yields state-of-the-art results for tabular data

all our code
is open-source:
github.com/automl

AutoML.org



Thank you for your attention!

Funding sources

BOSCH

erc European Research Council

Emmy Noether-Programm
DFG Deutsche Forschungsgemeinschaft

GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung

DFG Priority Programme 1527

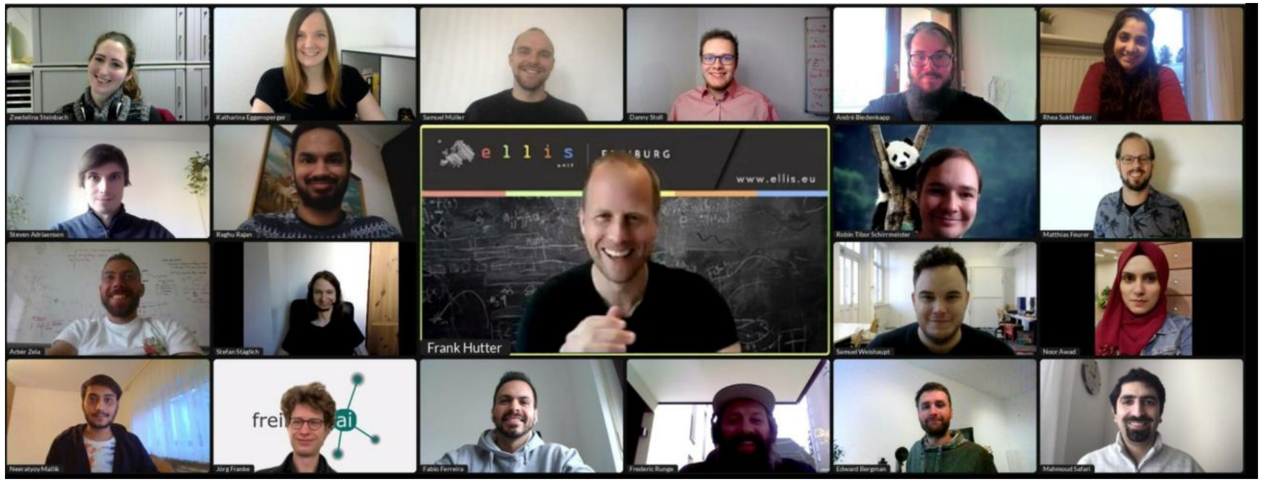
TAILOR https://tailor-network.eu/

DFG Deutsche Forschungsgemeinschaft

European Commission

BrainLinks BrainTools acting.thoughts

My fantastic team



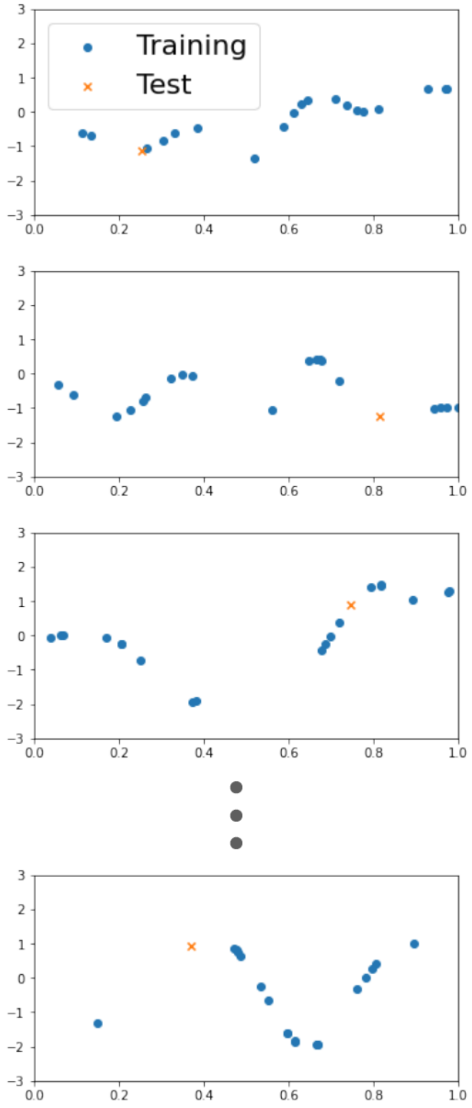
I'm always looking for new great team members
Please see automl.org -> jobs

@FrankRHutter

AutoML.org

Backup Slides

Illustration of Prior-Fitted Networks (PFNs) for Approximating GPs

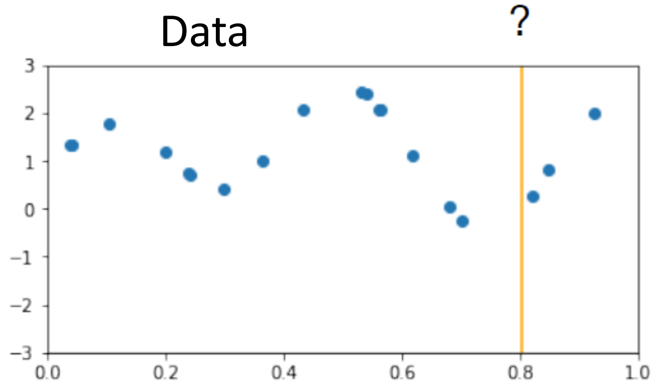


Samples from the prior

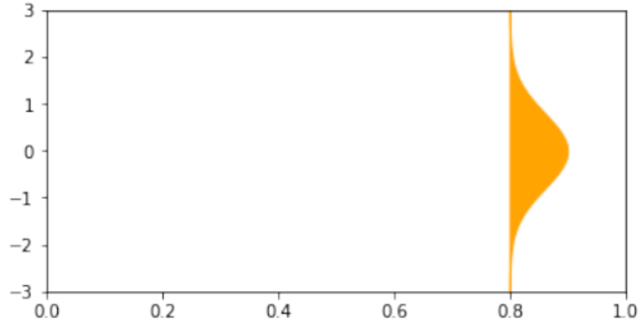
Learn a **model to predict test from train**

We call this model a **prior-fitted network (PFN)**

We have thus meta-learned to approximate Bayesian inference, purely by supervised learning with set-valued inputs



Perform a **forward pass with the PFN**



Posterior predictive distribution

Massive Progress in Speech Recognition in the Last Decade