IntroML

30 March 2022

IntroML

Welcome

The ML ⇌Science Colaboratory

Plan for the day

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Agenda

9:10 Round of introductions

1. **Core ML Concepts**
   ML & software ⁄ learning paradigms ⁄ supervised learning ⁄ NNs
2. **Machine Learning in Practice**
   project pipeline: data, models & evaluation ⁄ ML in Science
3. **Discussion of *Selected* Scientific Use Cases**
   question ⁄ learning task ⁄ dataset ⁄ evaluation

12:45 Closing – feedback round

Practical aspects

- Lunch veg stew/chicken chops: register?
- Pauses → 2 x 10’
- Internet → eduroam, share laptop
- Toilet → elevator
- Corona & ventilation

?! Questions welcome at any time

Slides, notebook and feedback at
mlcolab.org/**introml-participants**

Round of introductions

Tübingen is a great place to form a community of practice and mutual support around ML and science.

Let's get to know each other! We can share

- Name & affiliation
- Current research
- Expectations from ML related to your research

The ML ⇌ Science Colaboratory

At the Cluster ML in Science

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Mission **Establish ML** across disciplines in Tübingen

Activities **Cooperations** and **trainings**, **scientific ML software**

@sethaxen
Seth Axen

@elenasizana

@alpiges

@alvorithm
Álvaro Tejero

- **ML Colab** ⊂ Cluster ML in Science ⊂ ML Tübingen
- Scientists **like you**, only some subfields of ML
  Our domain languages
  Astro, **atmospheric**, & **quantum physics**,
  environmental sci, **neuroscience**, **genomics**,
  urban sci, int'l policy, **structural bio**…
- We can work together, more at the end.

Great Potential for ML in Research

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

\+ More data in sciences (sensors) and humanities (text understanding)

\- **ML is hard to use for domain researchers**

- **Resources —** abundant data and powerful compute
- **Planning —** coupled model-data development
- **Skills —** exploding ML literature, software engineering, training lore
- **Limitations for research —** biases, interpretation, uncertainty, and causality

\- **ML-to-domain *application gap*** Do research with *existing* ML algorithms?

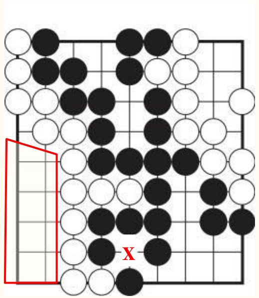
- Incentives misalignment in standard collaborations

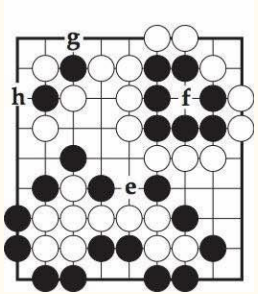
Core Concepts in Machine Learning

What is ML?

What problems does it solve?

Games and the nature of intelligence





- form territories by surrounding empty areas
- capture by completely surrounding (*prisoners*) **x**

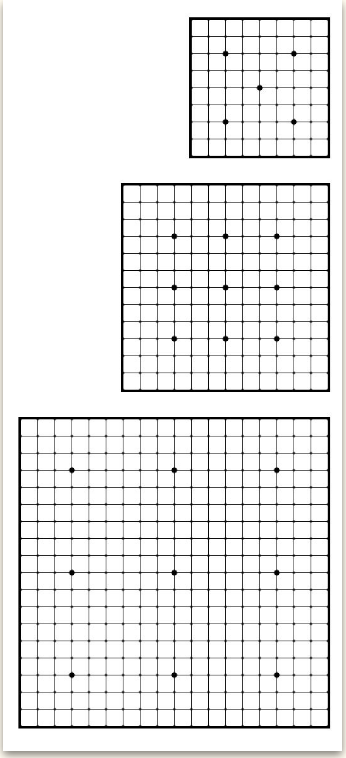
○ at **f** → **7** ● captured

○ at **g** or **h** → 1 ● each

● at **e** → 10 ○ and 10 crossings !

Points = crossings + prisoners

Real go is 19x19



British Go association

A surprising move

See also AlphaGo - The Movie

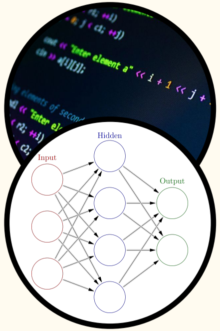
Creative?


ALPHAGO
BFI LONDON FILM FESTIVAL
WINNER
MILL VALLEY FILM FESTIVAL
Official Selection

Machine learning as ‘software 2.0’

- ML produces computer algorithms, called “ML models”
  - Like traditional computer programming based on **rules**, the result is computer code
  - But ML models are substantially based on **data**, not pre-specified rules
  - In fact, ML models *use* traditional code containing general **rules for learning specific rules**

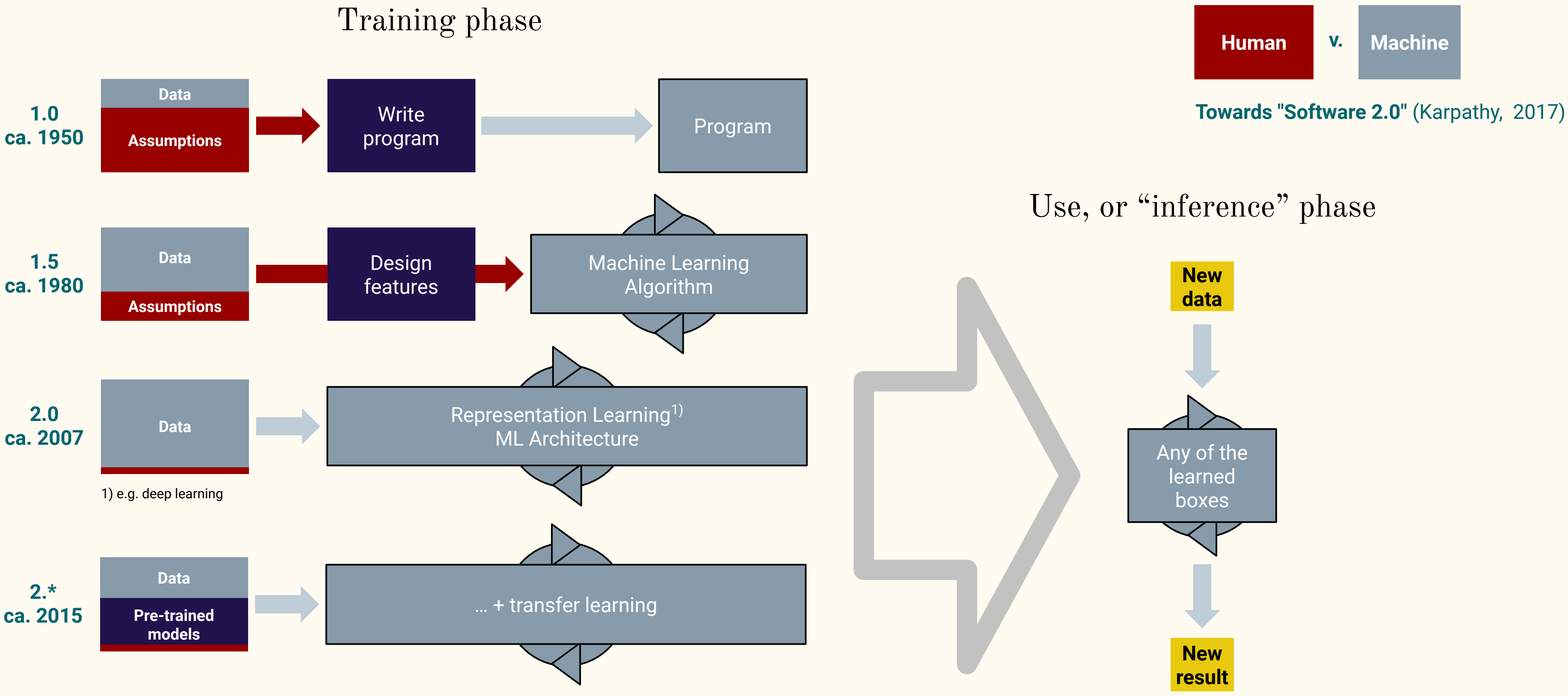
- Stages of an ML model and its internal parameters (IPs)
  - **Untrained** - it has not seen any data; IPs are typically random
  - **Training** - changing IPs progressively reflect presented data/experience
  - **Operational** - IPs fixed by training to values that satisfy a desired performance level



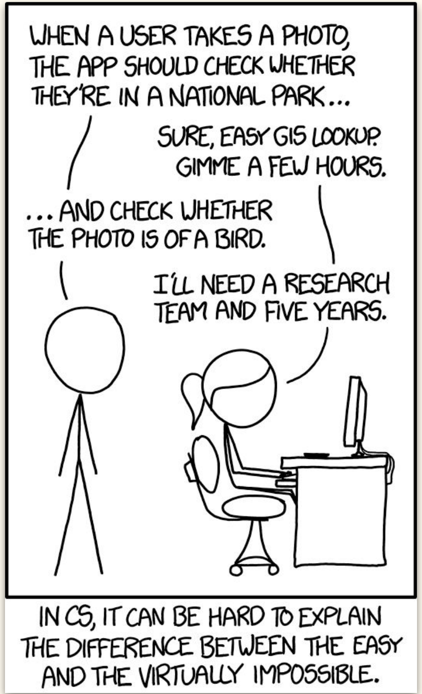
- Who interacts with ML when?
  - **ML researchers** design learning algorithms, train and test them, often on standardized tasks
  - **Users** train ML algorithms with their data for their *use case*
  - **Everybody** (almost) supplies data for 3rd-party ML, or uses ML trained by others

Karpathy 2017: Software 2.0, Jordan 2018: organizing ML projects

Towards Software 2.*



Different strengths



Use ML when the problem…

- … has a simple objective,
- is too complex for explicit rules,
- is constantly changing,
- is perceptual (“unstructured” data), or
- is observable, but unstudied due to scale e.g. network traffic logs

Don’t use if: every decision/change in behavior must be explainable; errors are high-consequence, getting dense data is infeasible or costly…

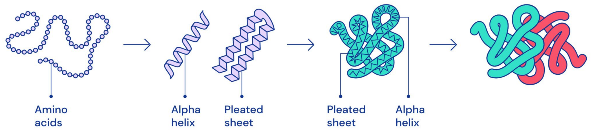
Why ML now?

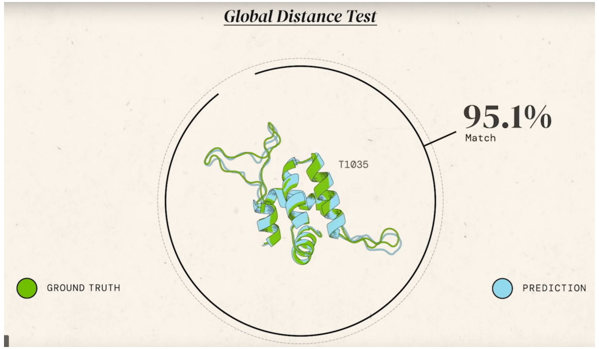
EBERHARD KARLS UNIVERSITÄT TÜBINGEN

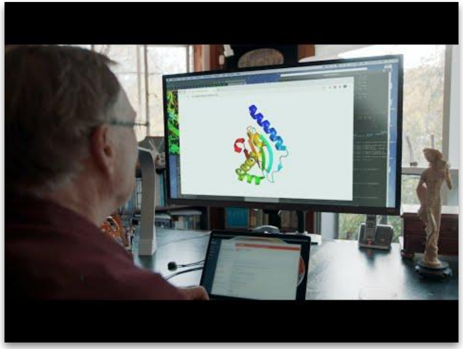
ML coming of age ca 2010, a coalition of factors

- Data: commoditization of sensors and distributed data collection
- Hardware: parallelized linear algebra, originally for gaming (“GPUs”)
- Software:
  - Usable automatic differentiation engines (e.g. Theano 2007)
  - Languages for composing models easily
- Models
  - Multiple layers and training tricks (grad clipping, dropout, batchnorm, …) to stabilize training
  - Compositionality of neural networks
  - Transfer learning

From proteins to planets, ML in science

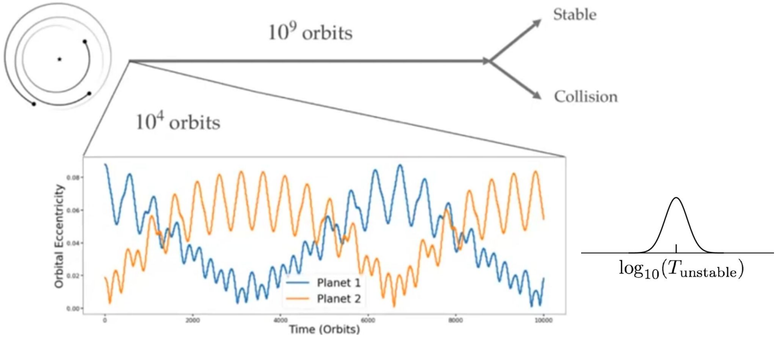






Protein structure prediction with transformers

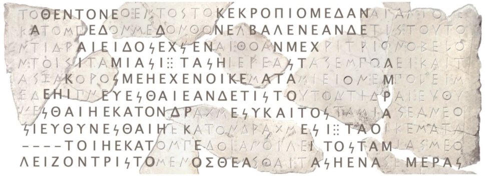
Senior et al. 2020, Jumper et al. 2021

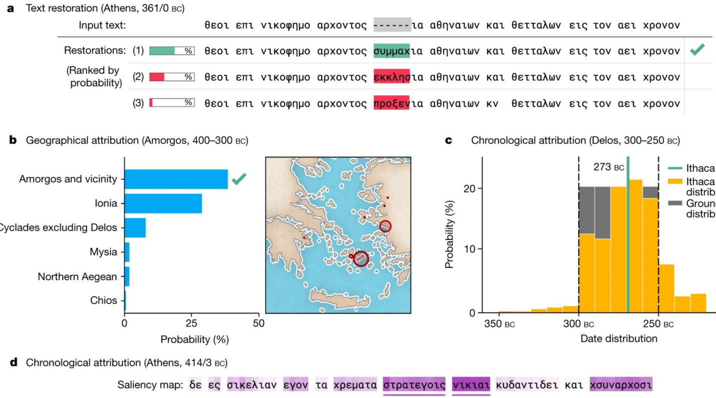


Planetary stability with Bayesian neural networks

Cranmer, Tamayo et al. 2021

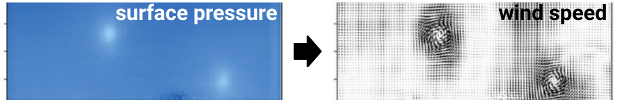
Ancient greeks and seasonal hurricanes

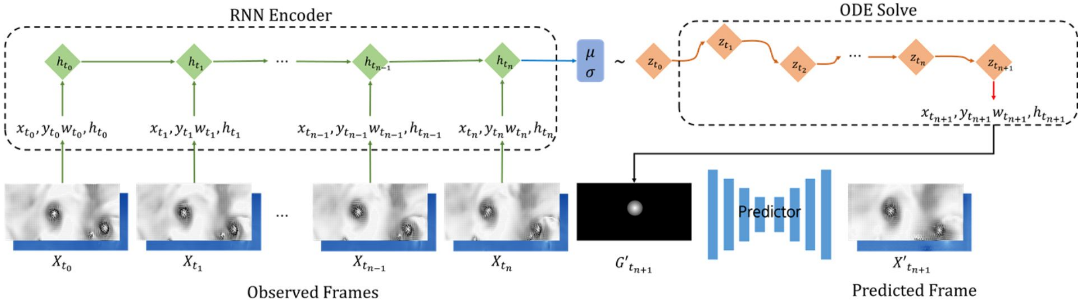




Text restoration and geochronological attribution with data augmentation

Assael, Sommerschield et al. 2022





Hurricane path forecast & speed field with nODEs & GANs

Park, Kim et al. 2020

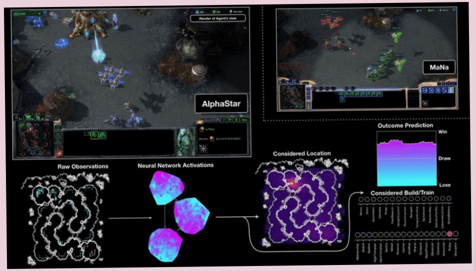
Core Concepts in Machine Learning

Learning Paradigms

Main Learning Paradigms

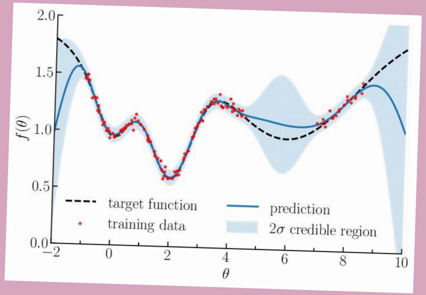
Reinforcement learning

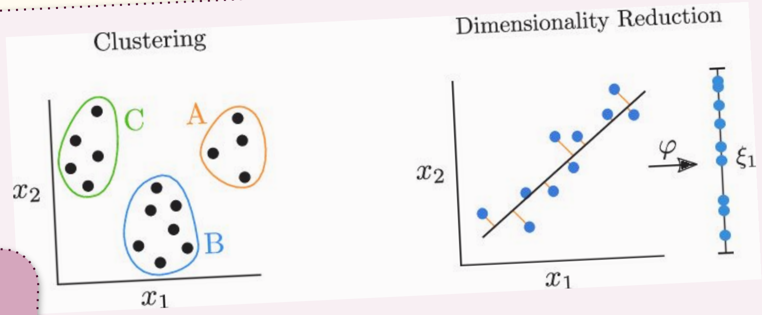
Harnessing interaction



Supervised learning

Teaching to label





Unsupervised learning

Exploring structure

Figures DeepMind 2019, Leclerc 2018, jovian.ai 2021

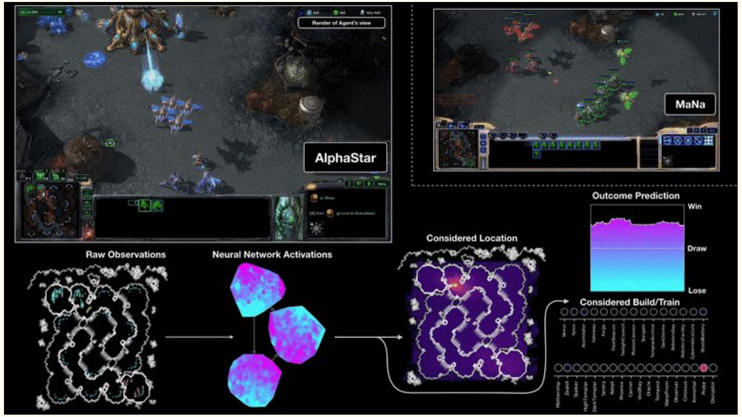
Reinforcement Learning

Goal

Train an agent to perform some task

Requirements

- Agent interacting with an environment (real or simulated)
- Reward mechanism

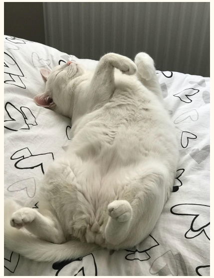
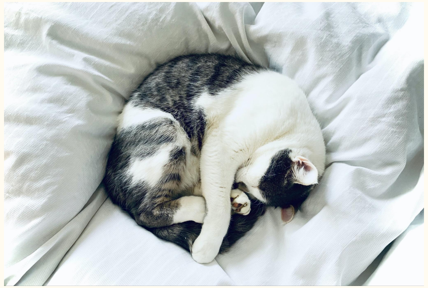
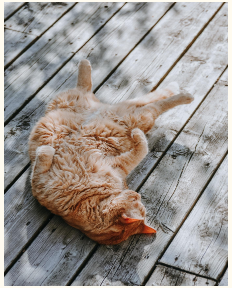


Reinforcement Learning

Reinforcement Learning

Unsupervised learning

An example dataset

Instance

ID	R01	G01	B01…
1	215	198	180 …
2	61	61	51 …
3	219	227	227 …
4	47	43	37 …

Feature

Photo by Sarah Ball on Unsplash Photo by Gökhan Konyalı on Unsplash Photo by Ben Wicks on Unsplash Photo by Jacalyn Beales on Unsplash

Finding structure in the data


Photo by Sarah Ball on Unsplash

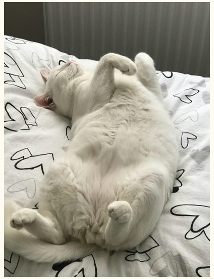


Photo by Gökhan Konyalı on Unsplash

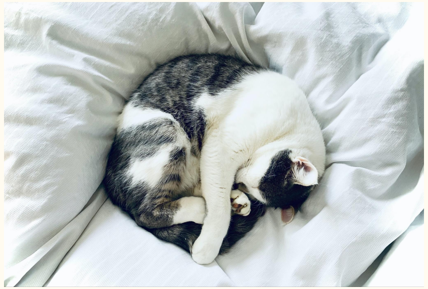


Photo by Ben Wicks on Unsplash

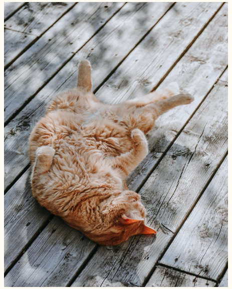


Photo by Jacalyn Beales on Unsplash

Task: group the images into any numbers of groups that you like

Finding structure in the data


Photo by Sarah Ball on Unsplash

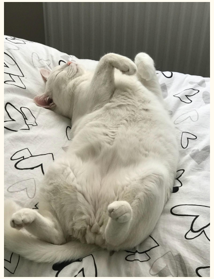


Photo by Gökhan Konyalı on Unsplash

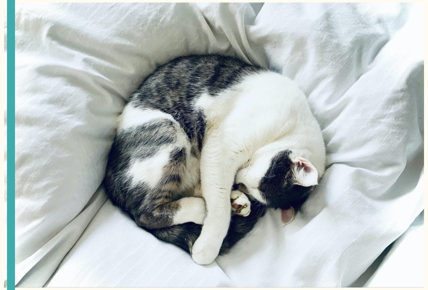


Photo by Ben Wicks on Unsplash

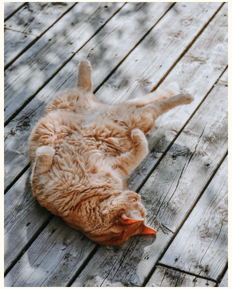


Photo by Jacalyn Beales on Unsplash

For example: 4 groups based on the breed of cat

Finding structure in the data

Photo by Sarah Ball on Unsplash

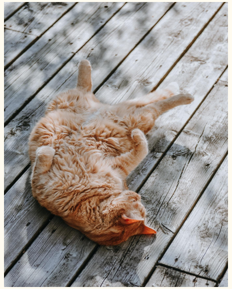


Photo by Jacalyn Beales on Unsplash

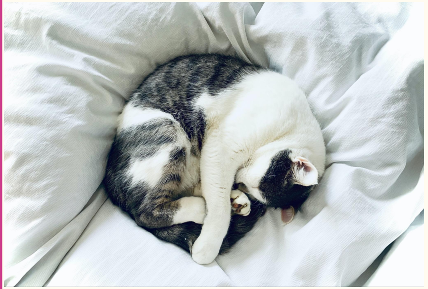


Photo by Ben Wicks on Unsplash

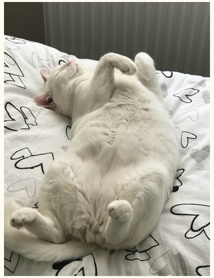


Photo by Gökhan Konyalı on Unsplash

For example: 2 groups based on colors (orange or not orange)

Finding structure in the data

Photo by Sarah Ball on Unsplash

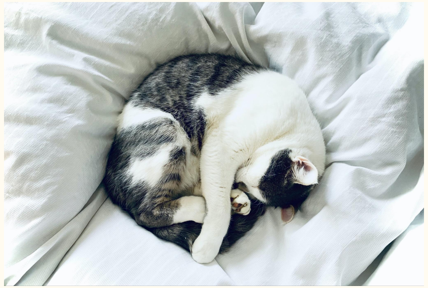


Photo by Ben Wicks on Unsplash

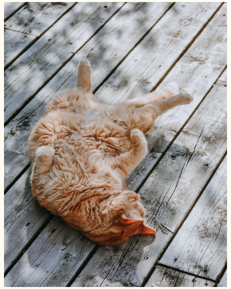


Photo by Jacalyn Beales on Unsplash

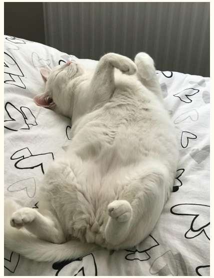


Photo by Gökhan Konyalı on Unsplash

For example: 2 groups based on gesture

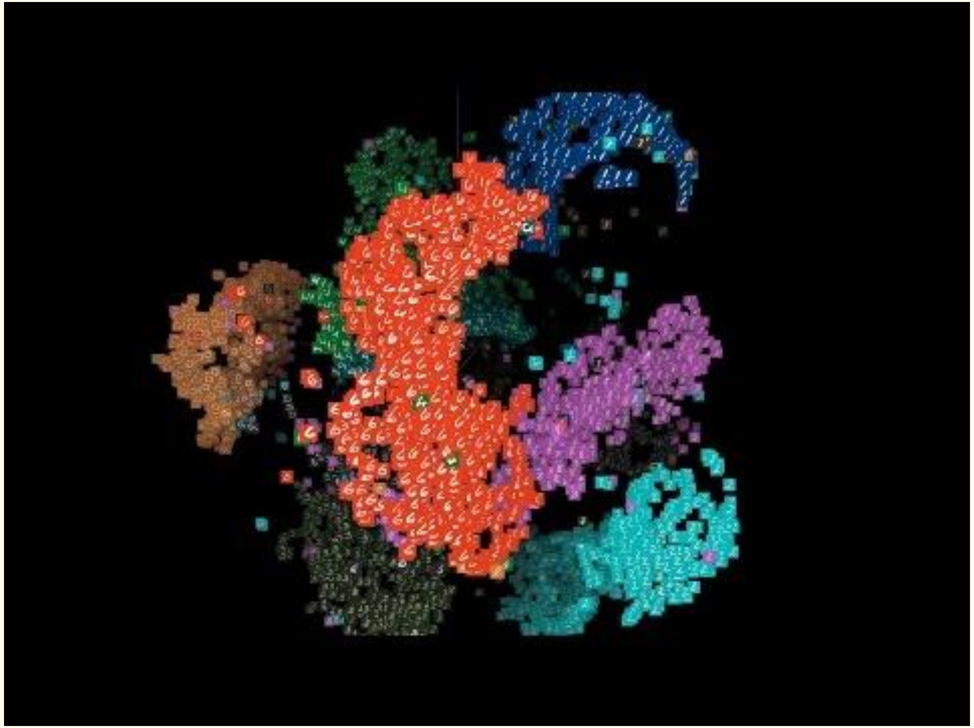
Unsupervised Learning

Goals

- Structure discovery
- Dimensionality reduction
- Community detection

Requirements

- Data (unlabeled)
- Some notion of similarity of data points



Core Concepts in Machine Learning

Supervised learning

One step at a time

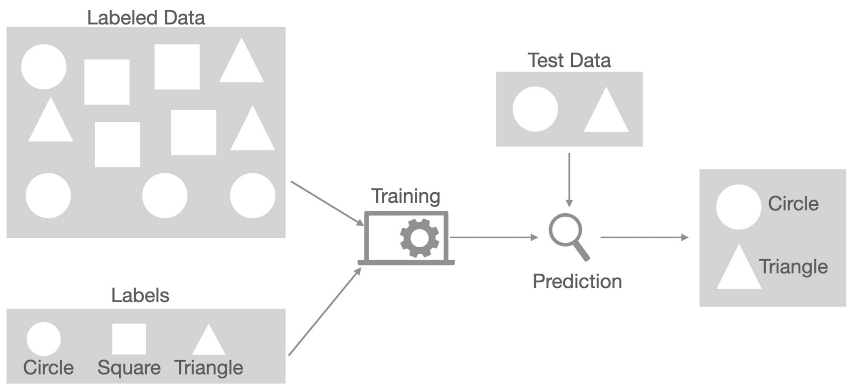
Supervised Learning

Goal
learn a function to map inputs to outputs

Requirement
Dataset of inputs *and outputs* ("labels")

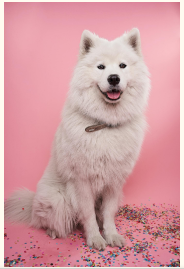
Modes

- Classification (category outputs)
- Regression (continuous outputs)

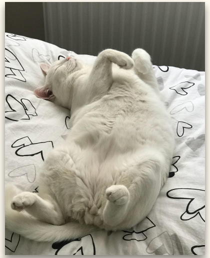


Supervised Learning: Classification

Cat



Dog



Cat

Dog

Category Label

ID	R01	G01	B01…	Type
1	184	187	187 …	Cat
2	218	157	164 …	Dog
3	61	61	51 …	Cat
4	236	236	236 …	Dog

Supervised Learning: Regression

47

0

2

-33

Continuous Label

ID	R01	G01	B01…	Head pose (degree)
1	116	98	81…	47
2	157	146	164 …	0
3	23	22	21…	2
4	137	130	129 …	-33

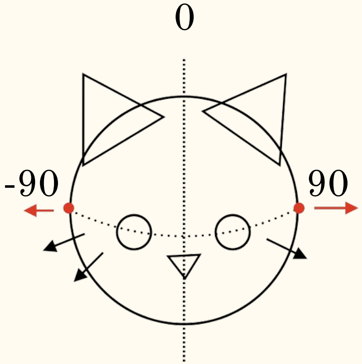
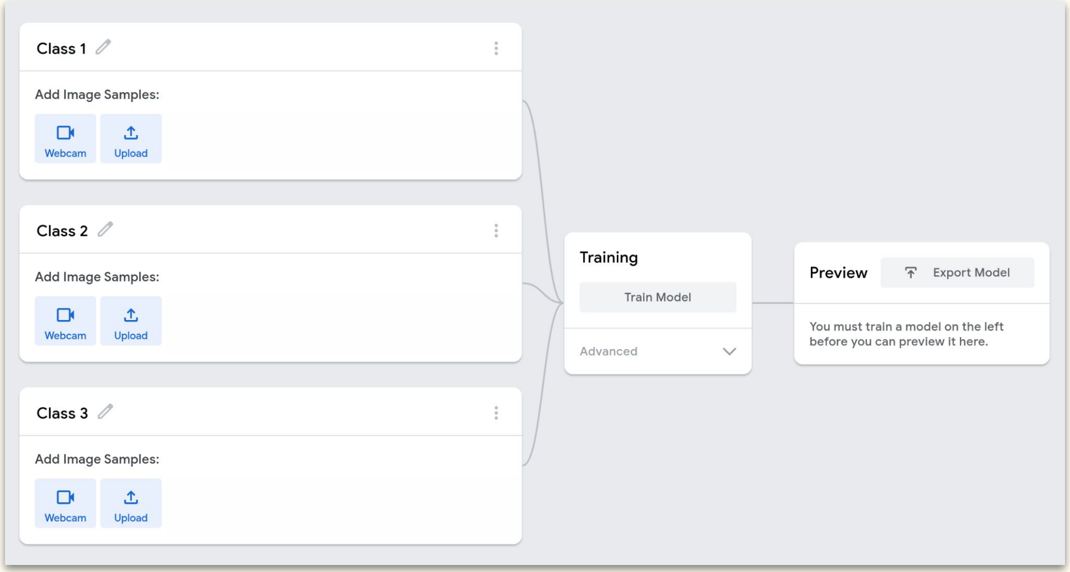


Photo by: cat-breeds-classifier

Let's train a supervised classifier



Navigate to:
https://teachablemachine.withgoogle.com/train/image

We’ve trained a supervised classifier

- Collect data
  image frames + labels
- Train*
  neural network
  * fine-tune
- Infer or predict
  class of new image

ML brings the senses to computers

Supervised Learning Tutorial

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Supervised learning: One step at a time

In this notebook, we slowly introduce supervised learning, along with basic machine learning concepts we encounter on the way.

The data

At regularly spaced 1-dimensional points x, we have generated fake, noisy 1-dimensional observations y. We assume that there is some true but unknown underlying process f, so that

$$y_i = f(x_i) + \text{noise}.$$

x_i is a single **raw feature**. y_i is an **output** or continuous **label**.

Interactive notebook downloadable at mlcolab.org/introml-participants

Core Concepts in Machine Learning

Neural Networks

Layer upon layer upon layer

Building a neural network - playground

Visit a simplified version or go to https://playground.tensorflow.org for the general interface

- **Input data**
  - 'DATA' *chooser*: blobs, quadrants, concentric & spiral
  - 'OUTPUT' *visualizer*
    - Raw features: coordinates of the dots, X_1 and X_2
    - Supervised labels: binary classes
    - Training data; tick checkbox for test data
- **Model**
  - 'FEATURES' add *computed features* X_1^2 , X_2^2, X_1X_2, sin X_1, sin X_2 to form an *input layer*
  - 'HIDDEN LAYERS' select *architecture* by adding fully connected *hidden layers* of desired size.
- **Output**
  - 'OUTPUT': f(X_1, X_2, *features*(X_1, X_2)) as function of X_1, X_2 shown as color level
  - Input dots overlaid for visual comparison.

Training the neural network

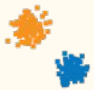
Training

- → Start: initialize *weights* randomly, (hence *activations* also random)
- □ Learn: blame training loss on specific weights, and adjust them to reduce it
- □ Stop: see when training loss (or test error) plateaus, call it a day

Inference and evaluation

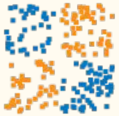
- Check on a separate test set how well we generalize
    - E.g. do all spirals have arms this long? Distance between the arms?

Challenge time!



blobs

concentric



xor

spiral

blobs: find simplest network, using raw features only

concentric: same, then find simplest with engineered features

xor: simplest network *with* engineered features, then without

spiral: 1 layer and engineered features vs. deep and raw features only. ReLU

Think about:
How you find/invent features? how/when to trim the network? how good is extrapolation? why oscillations in the loss? why different results every time?

Machine Learning in Practice

The ML ⇌Science Colaboratory

Machine Learning in Practice

ML development cycle

From data to scientific results

Zooming out: the ML dev cycle

Let's talk about how to…

1. … define the task
2. … obtain data, and prepare it
3. … implement the model
4. … evaluate, interpret and report output

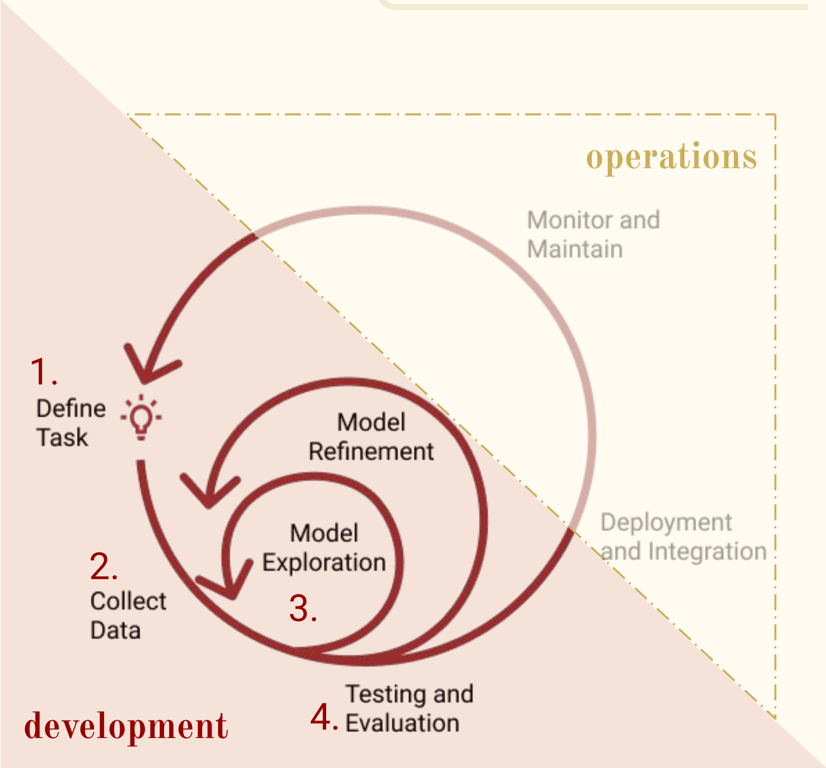
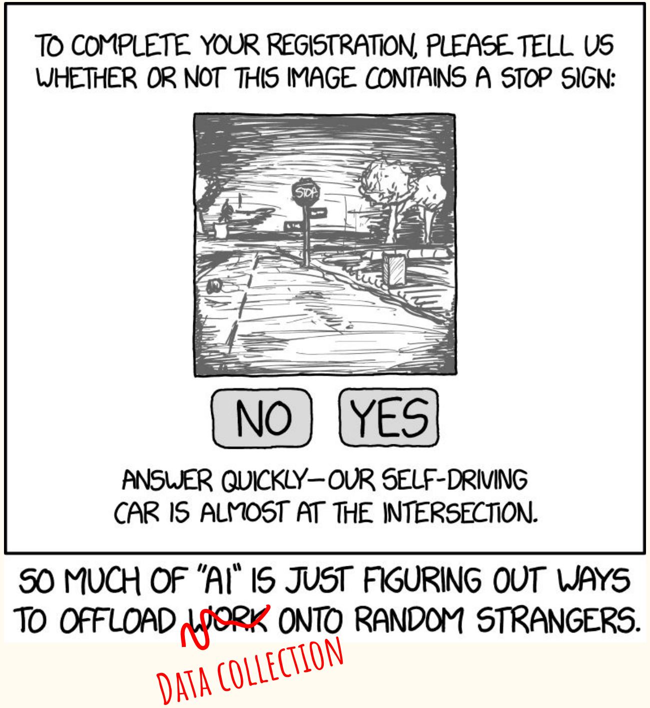


Figure based on Jordan 2018: organizing ML projects

0. Task: Feasibility & Impact of ML Project

- Can performance be measured?
- Can humans do it? With what performance?
- Does the data cover all relevant regimes/conditions? (**density**, not size!)
- How much does additional (labeled) data cost?
- What is the consequence of wrong predictions? (false alarms and omissions)
- Does it need to perform well to be useful?
- Does it need to be interpretable?
- Which are performance baselines in your field? (even if non-ML)
- Are there pre-trained models you can leverage? (esp. NLP/foundation models)
- Is the task really necessary, are there shortcuts?

1. Data: How to get It



xkcd.com/1897

Strategies

- **Reduce** via pretrained models
- **Find** public datasets in your field
- **Complement** with public ancillary data
- **Augment** with computational transforms

Sources

- Your own *sensor* data. Cheap!
- Your own *survey* data
  - Passive data collection? (need to provide value)
- Labeling
  - Specialized user interface
  - Active learning
  - 3rd party services: gamification
  - Cross-human agreement metrics!

1. Data: Common Modalities

- Time series
- Images
- Relational (graph) data
- Tabular
- Video
- Georeferenced measurements
- Spectra
- A mix!

Data modalities go hand in hand with a specific approach to reflect aspects such as

- The future depends on the past
- Close points are usually similar
- The curvature of the Earth matters

Icons from Icons8

1. Data: Pitfalls

- Systematic error (bias)
- Contradiction (e.g. databases with different conventions)
- Noise (i.e. “random” error)
- Independence across samples (e.g. are errors/noise correlated?)
- Missingness
  - Completely at random (missingness independent of known or unknown features)
  - At random (missingness could be predicted from known features)
  - Not at random (missingness could not be predicted from known features)
- Size/resolution (e.g. images with different dimensions/DPI)
- Multi-scale phenomena
- Censored data (data thresholded to some interval, e.g. bathroom scale)

1. Data: Processing

- Every dataset has a story; being able to explain that story is important for ML.

- As a domain expert, you know a lot about the data that you might take for granted

- Examples:
  - pollen counts
  - outlier removal

2. Model: Training, Validating and Testing

- **Training** for **best parameters:** training data ▮▮▯, minimize loss function

- **Validation** for **best hyperparameters:** held-out data ▯▯▮, evaluate metrics for model settings that are not differentiable
  - Examples: # layers, # units, learning rate, activation function, rescaling of inputs ...
  - All combinations (=”grid search”) quickly impracticable.
  - Smarter: random sampling, Bayesian optimization, dataset-size extrapolation, …

- **Test** for **generalization performance:** test data ▯▯▯ ▮🔒, report metrics
  - Compare with **baselines** (random or specific) and toplines (human performance)
  - Monitoring: evaluate on fresh data to check for degradation:
    - relevant for operational systems,
    - caused by covariate shift, target shift or concept drift.

2. Model: Implementation & Embodiments

- **"Classical ML"**
  - stable, documented, interpretable
  - CPU arrays (numpy, R)
  - R, sklearn, mlj
- **Probabilistic programming**
  - complex models
  - Bayesian outputs
  - model checking culture

  - CPU mostly, expensive
  - Sampling + AD + probabilistic DSL
  - PyMC, Pyro, Stan, turing
- **Deep learning**
  - expressive, versatile, hard to interpret
  - Training vs inference
    - Resources & devices
  - AD + arrays on the GPU + computational kernels + layering syntax
  - PyTorch, TensorFlow, jax, flux
- **Pretrained DL (that you *can't* train)**
  - API serving (per-query cost)
  - or *download-and-finetune*
  - Huggingface, GPT, ...

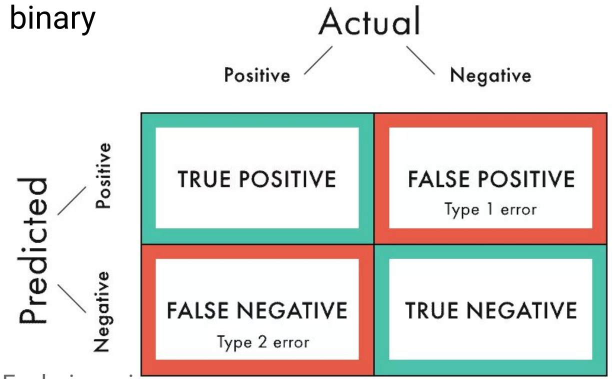
3. Evaluation & Performance Metrics

- Use held-out test data 🔒 to report & compare models
  - *Benchmarks* are combinations of task + dataset used to drive development in ML
- You get what you optimize for…
- … but not all you care about we can optimize for → need metrics, not just loss
- Metrics for:
  - Regression: MSE (cf. poly regression), MdAE (robust), MAE, …
  - Classification: precision, recall, F-score, (cost-sensitive) accuracy, Cohen's **κ** …
  - Ranking, e.g. search: several metrics that weight more correct top results

3. Evaluation: Classification Metrics

- False alarms (FPs) vs. omissions (FNs)
- ML → decisions, can't ignore **asymmetric losses**, e.g. early diagnostic w. $L_{FP} << L_{FN}$
- Know **your goals (error penalties)** *and* **your data (class imbalance)**
- Key concepts: precision, recall, F_β-score, P-R curve, ROC (TPR-FPR) curve, AUC.

binary



Explorium.ai

5-class

Perceived vowel / Vowel produced	i	e	a	o	u
i	**15**		1		
e	1		1		
a			**79**	5	
o			4	**15**	3
u				2	**2**

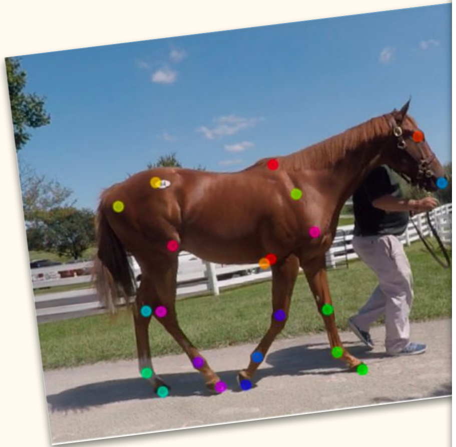
Wikipedia

Machine Learning in Science

Learning from language, images, simulations…

Computer Vision (CV)

Data: images or videos



Pose estimation

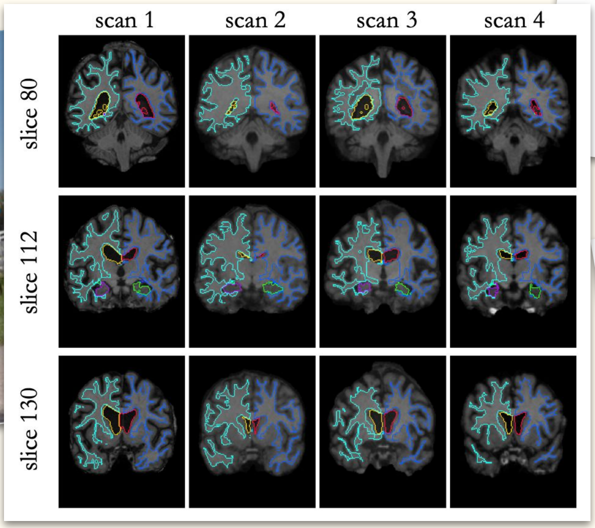
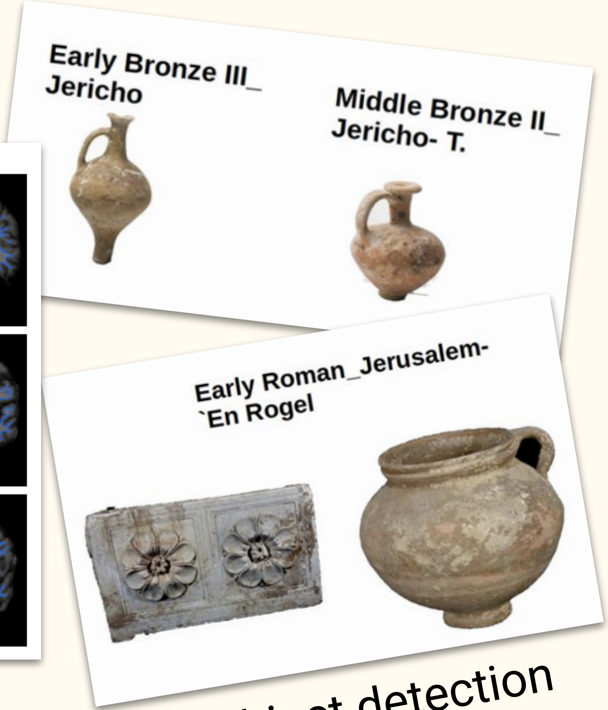


Image segmentation



Object detection

Balakrishnan, Zhao et al. 2019,
Nath, Mathis et al.2018, Resler, Yeshurun et al. 2021

CV: pose estimation without body markers

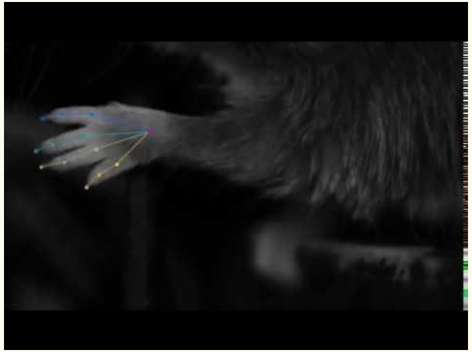
Hypothesis We can detect body parts of multiple animal species from unstaged video without body markers

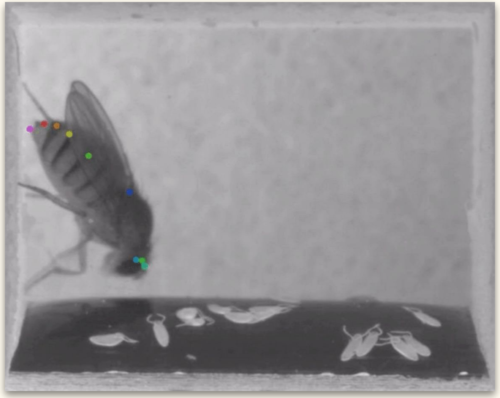
Dataset video frames with *labeled* positions of body parts

Task frame-based pose estimation

Evaluation *RMSE* between predicted and labeled positions

Model *pre-trained* ResNet with fully convolutional upsampling

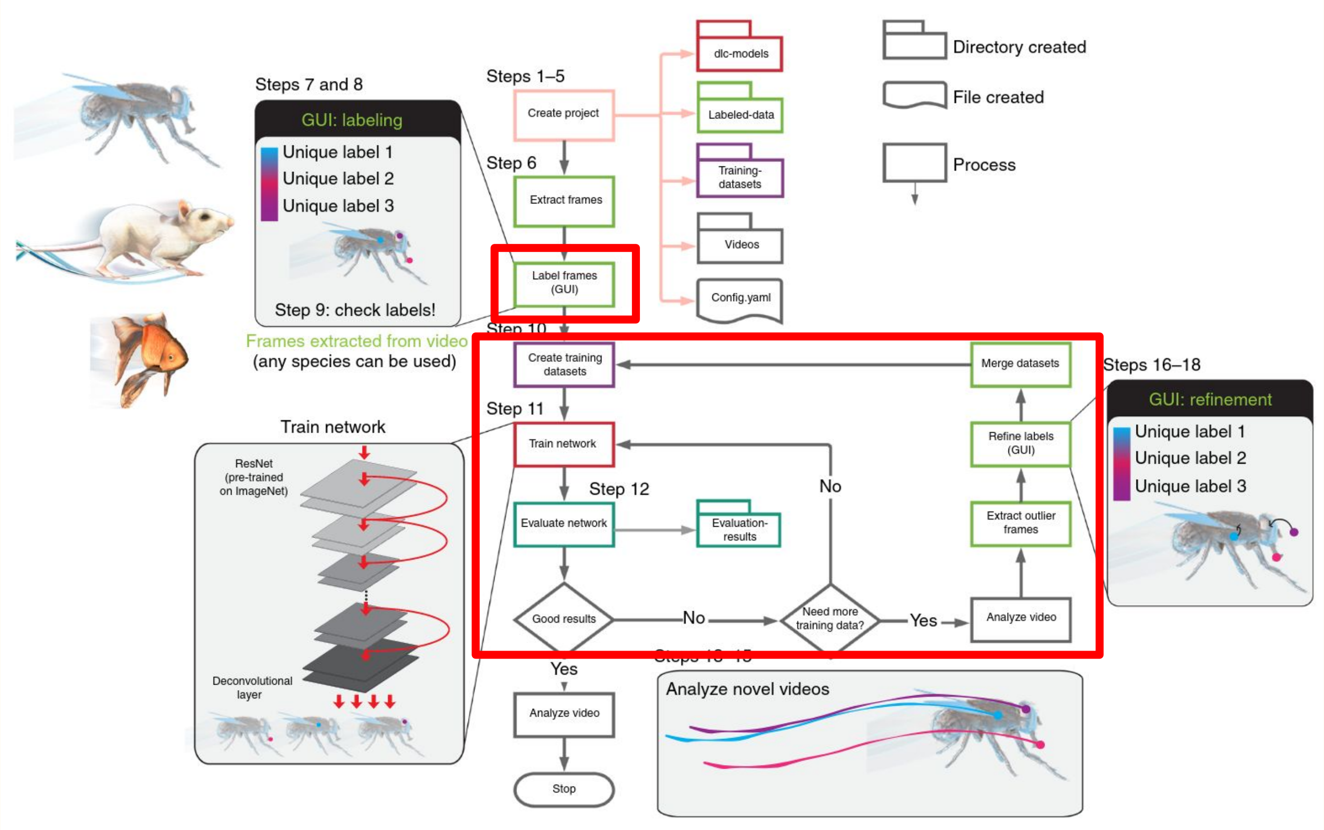




Fruit fly

Nath, Mathis et al.2018

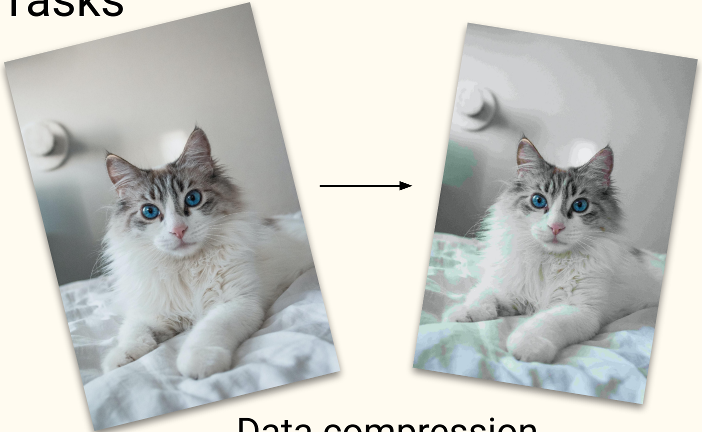
CV: pose estimation without body markers



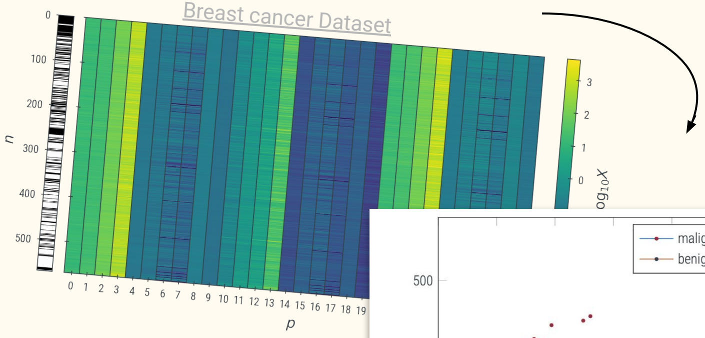
Nath, Mathis et al.2018

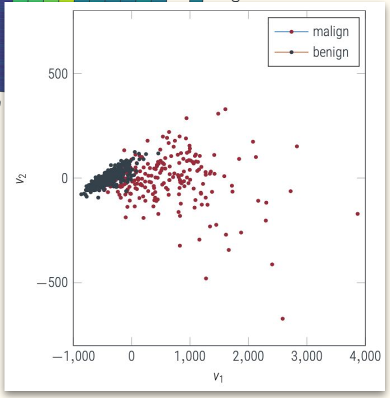
Dimensionality reduction

- Data
  - High-dimensional data (e.g. k space data)
- Tasks



Data compression



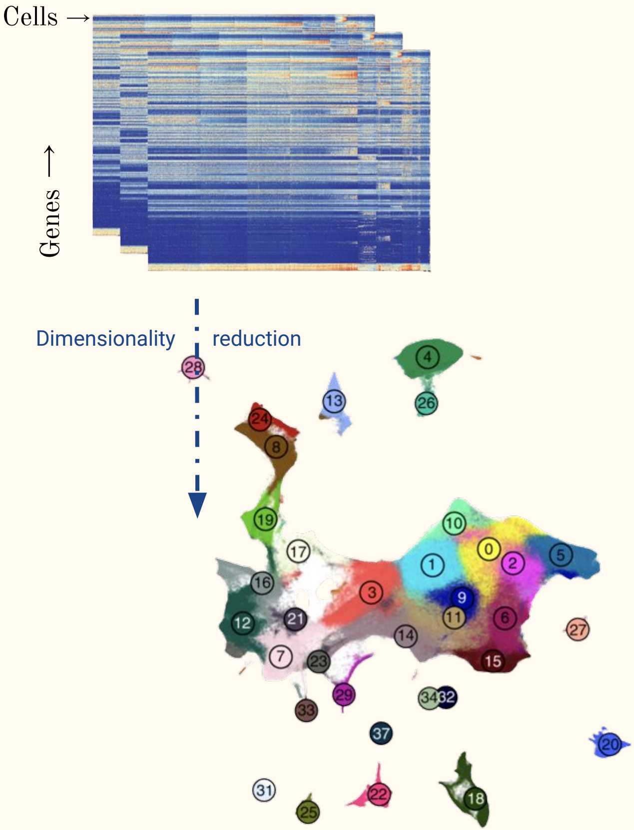


Visualization of high-dimensional data

- Models
  - Unsupervised models
    - Principal component analysis (PCA) & Kernel PCA
    - Autoencoders
    - t-SNE, UMAP (for visualization)
- NOTE: Often for pre-processing of the data

Cell-type prediction for single-cell transcriptomics

Luecken,Theis et al.2019



Hypothesis we can identify cells with similar types using their gene expression profiles

Dataset gene expression matrix (matrices of read counts per million)

Task *dimensionality reduction*

Evaluation fraction of neighbor cells preserved in the lower dimensional space

Model *t-SNE* with PCA initialisation

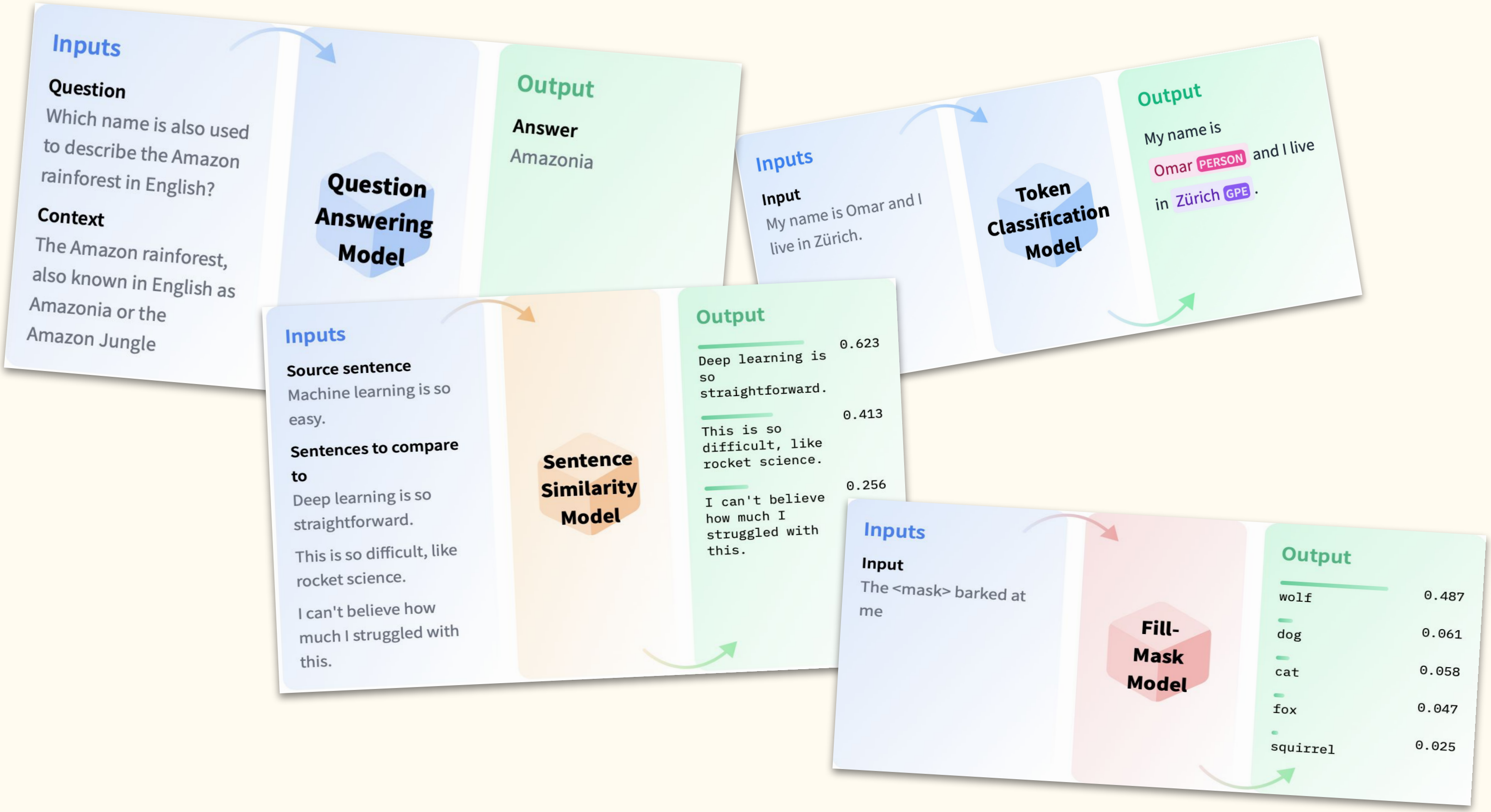
Kobak, Berens et al.2019

Natural Language Processing: tasks

huggingface.co: tasks

NLP: Tasks

EBERHARD KARLS UNIVERSITÄT TÜBINGEN



huggingface

NLP: Author Attribution in Latin

Hypothesis rhythmic constructions in Latin help reveal author identity

Data *Samples*: ~37k prose fragments of 10 consecutive > 4-word sentences. *Labels*: Author class
Features (computed) Baseline stylometric features, topic independent
Syllabic length (SL): ∪ short, — long, X anceps.

Arma vi|rumque ca|nō, Trō|iae quī| prīmus ab| ōrīs|| → – ∪ ∪| – ∪ ∪ | – –| – –| – ∪ ∪ | – X||

Task author attribution = fragment *classification*

Evaluation cross-entropy *loss* for training, *F*-score M/m *metric* for early stopping & feature importance

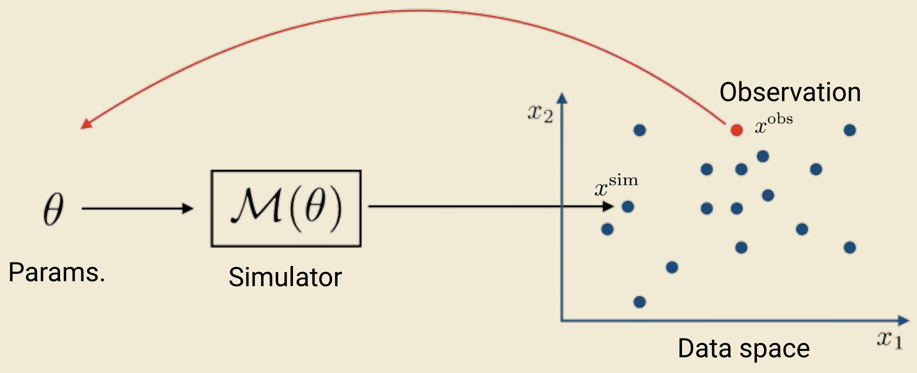
Model multi-channel NN (BF, SL, DVs), CharCNN 5 layers, avg proba decision

Corbara 2021 (preprint)

Simulation-based inference (SBI)

Tasks

- General inference in absence of likelihoods (losses)
- Particularly suited for black-box simulators $x^{sim} = \mathbf{M}(\theta)$.
- How can we learn a distribution over the parameters θ such that $\mathbf{M}(\theta) \approx x^{obs}$



Which parameters can reproduce the observation?

Note

We have a full 3-day workshop on SBI

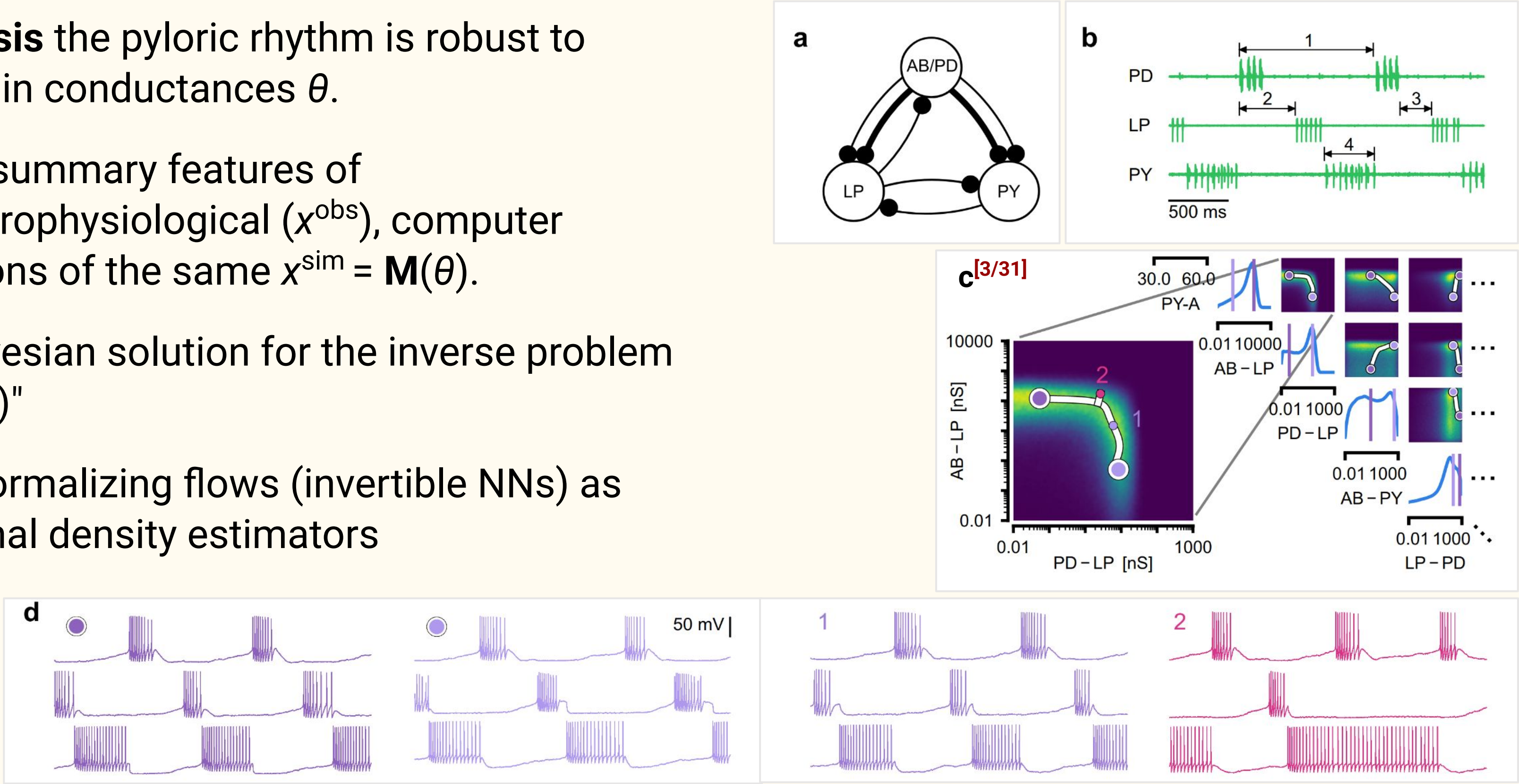
SBI: Stable Firing in the Pyloric Ganglia

Hypothesis the pyloric rhythm is robust to changes in conductances θ.

Dataset summary features of cell-electrophysiological (x^{obs}), computer simulations of the same $x^{sim} = \mathbf{M}(\theta)$.

Task Bayesian solution for the inverse problem "$\theta=\theta(x^{obs})$"

Model normalizing flows (invertible NNs) as conditional density estimators



[Gonçalves et al. 2020](#)

Limitations of Today's ML for Science

- Inductive paradigm, limited by data
  - Bad at extrapolation
- Building in inductive biases still a craft
- Compute intensive, environmentally questionable
- Correlational, mostly not causal
  - But can help you fit mechanistic models that are causal
- Interpretability of expressive models mostly post-hoc
  - Model-specific and model-agnostic techniques
  - Global and local explanations
  - Contrastive explanations
  - Feature importances
  - Much ongoing research!

Machine Learning in Science

Bring Your Science Problem

Task, data, metrics…

Checklist for discussion

- What is the scientific question?
- Describe your dataset: data modalities, labels (if there are), size
- Which part of the work might involve machine learning?
  - What type? Supervised, unsupervised, …?
- What is 'success'? How do you measure it?
- Any relevant baselines? toplines?

The ML ⇌ Science Colaboratory

Connecting ML with Science

Our Activities

- For Tübingen *domain scientists*
  - Advice on the use of ML
  - Cooperations spanning a few months (you already filled the form!)
  - Trainings
- For Cluster *ML researchers*
  - Matching with domain researchers
    - Interesting problems and new datasets,
  - Increase applied impact of research projects (e.g. PhD innovation fund: structured learning)
- For the *community*
  - Sharing best practices in applied ML
  - Developing software to democratize access to ML
  - Nurturing a welcoming culture for women in applied ML

Mapping requests to actions

Scientific *Infrastr.*
Cooperation
We execute and provide resources

Joint application
Colab & partner get resources

Ongoing consultation
We advise team, do not execute

Specific advice ▶ **advice & training actions**
Ocassional consultation

Mapping requests to actions

Data

Research question

Skills & resources (partner)

MLColab ⇌ partner fit

Scientific Infrastr. **Cooperation**	~~**Joint application**~~	**Ongoing consultation**	**Specific advice** ▶ **advice & training actions**
We execute and provide resources	Colab & partner get resources	We advise team, do not execute	Ocassional consultation

The Methods Center – *Methodenzentrum*

An institute for quantitative analysis **in the social sciences**.

Expertise

- multivariate & multilevel methods,
- (nonlinear) latent variable models,
- item-response theory,
- structural equation models,
- longitudinal studies, and
- causal mediator models.

Services

- consulting
- cooperations
  - publications
  - joint applications.

✉ office@mz.uni-tuebingen.de

In closing...

Thank you very much for attending our first IntroML **workshop!**


Seth


Álvaro


Elena


Alex


Yutong

One last thing,

… we thrive on critical, constructive feedback

please visit **mlcolab.org/introml-participants** and give us some!