CENTER FOR SCALABLE DATA ANALYTICS AND ARTIFICIAL INTELLIGENCE

Introduction to Machine Learning

Robert Haase

Funded by

Bundesministerium für Bildung und Forschung

SACHSEN

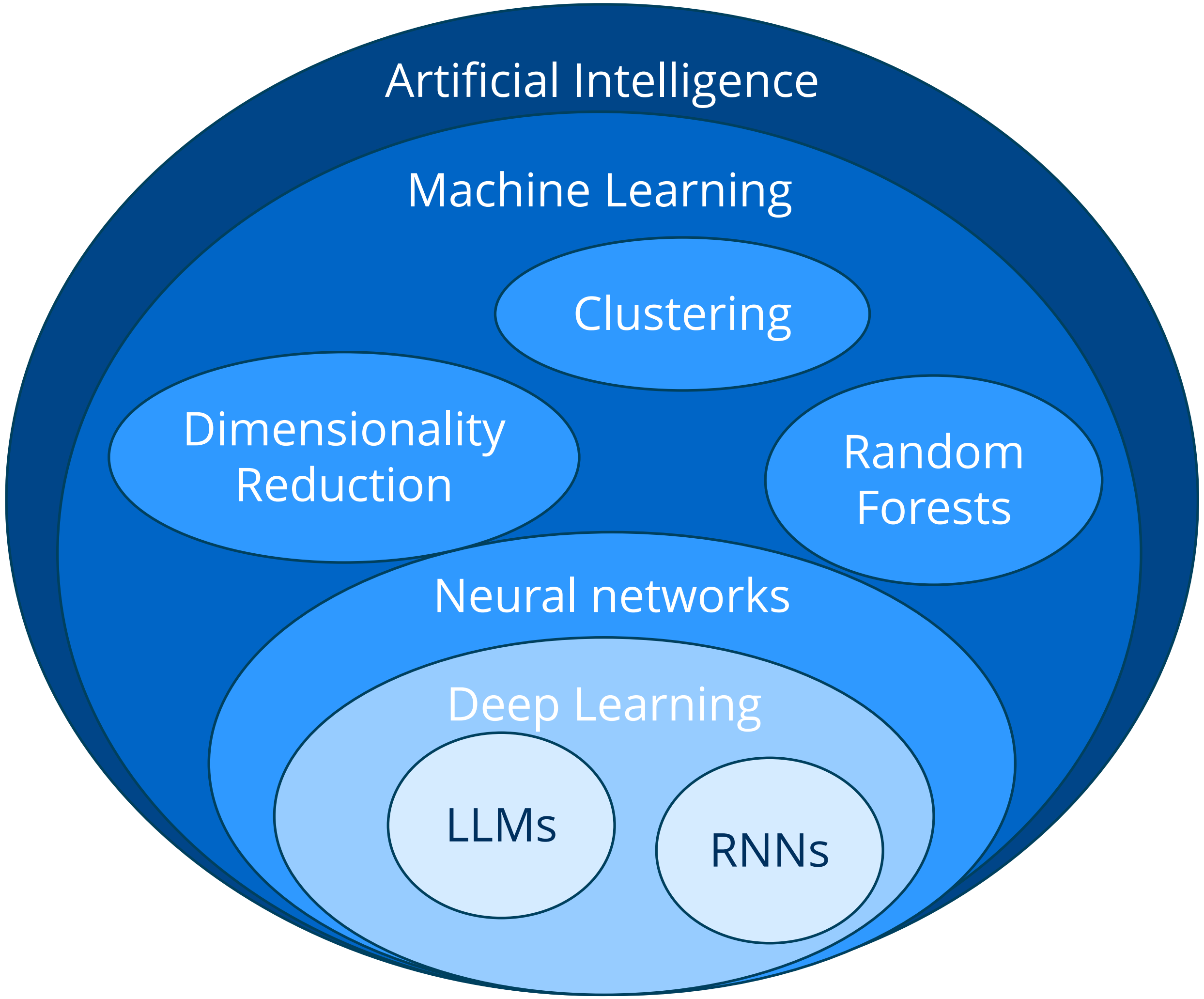
Diese Maßnahme wird gefördert durch die Bundesregierung aufgrund eines Beschlusses des Deutschen Bundestages. Diese Maßnahme wird mitfinanziert durch Steuermittel auf der Grundlage des von den Abgeordneten des Sächsischen Landtags beschlossenen Haushaltes.

Reusing materials from Johannes Soltwedel, Till Korten, Johannes Müller, Laura Žiguty (TU Dresden), Ryan Savill (MPI-CBG), Matthias Täschner (ScaDS.AI/Uni Leipzig) and the Scikit-learn community.

TECHNISCHE UNIVERSITÄT DRESDEN

UNIVERSITÄT LEIPZIG

Artificial intelligence



Artificial intelligence

Narrow AI
- Application specific
- Trained on labelled data
- Reflexive tasks
- Cannot extrapolate

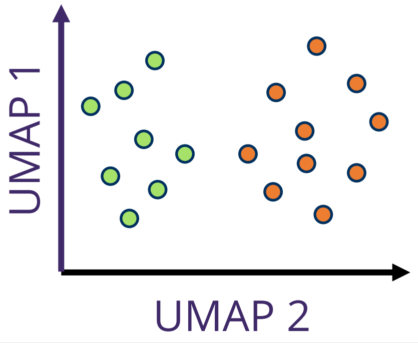
Great for data analysis tasks

General AI
- Human capabilities
- Access to knowledge of humanity, beyond individuals
- Can create *new* solutions by working creatively

Artificial intelligence

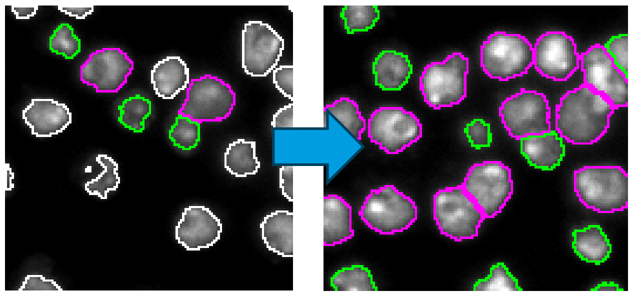
Unsupervised ML

- Dimensionality reduction
- Clustering
- Detecting patterns in unlabeled data
- Hypothesis generation



Supervised ML

- Learning tasks otherwise only humans could do
- Train a model, predict a classification



Generative AI

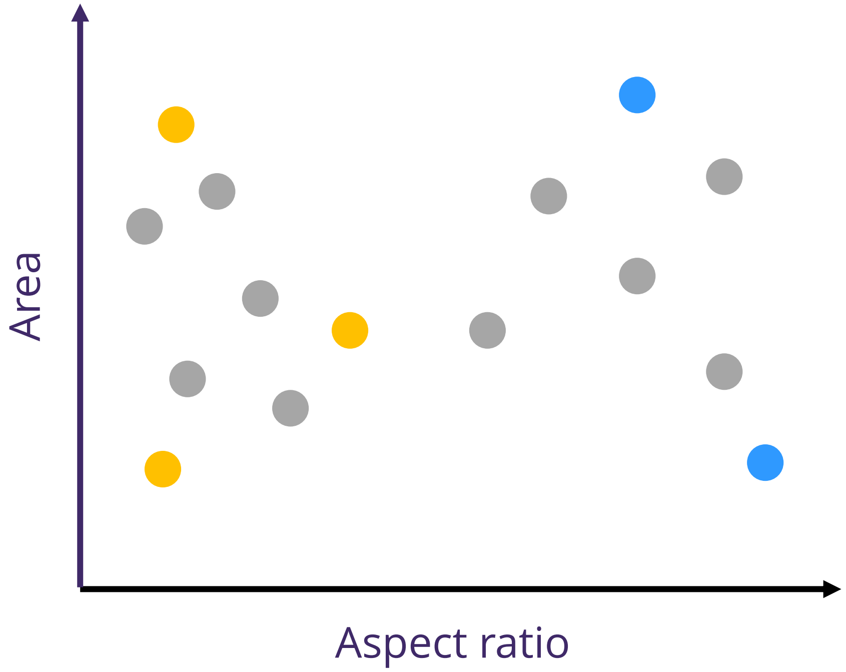
- Produces new data provided a context, often with human language prompts
- Hyped since 2022, with yet unclear limitations

Labelled data

- E.g. for shape differentiation of objects
- Partially labelled data

Bias?

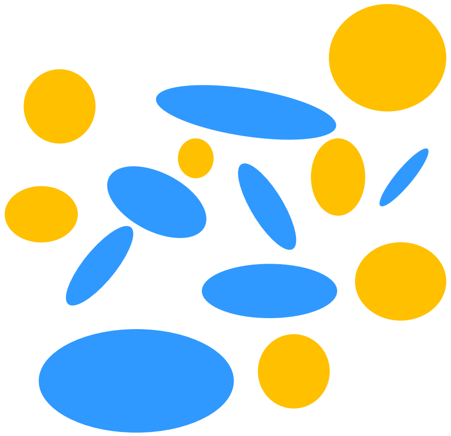
Elongated
Round
Unlabelled



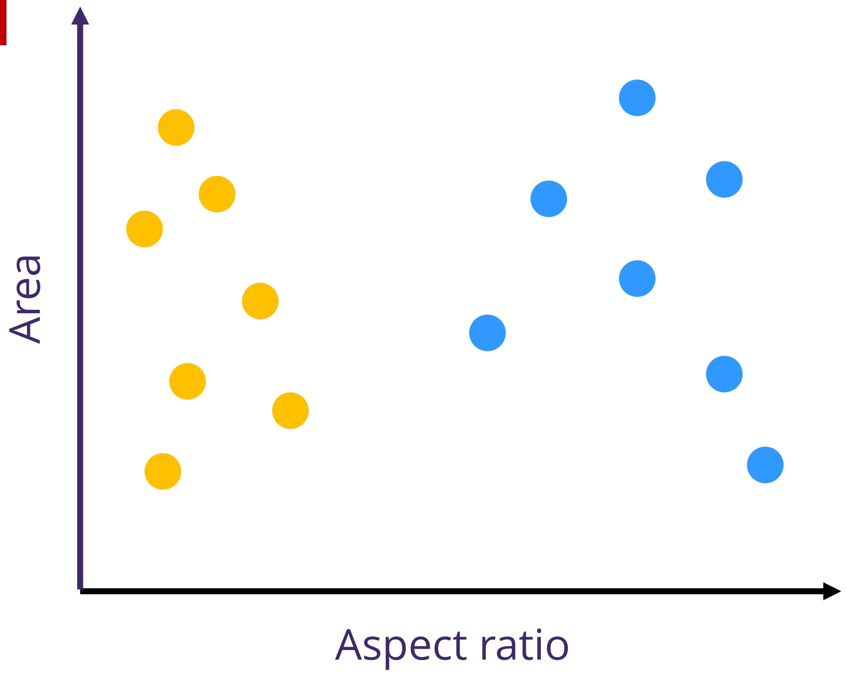
Labelled data

- E.g. for shape differentiation of objects
- Fully labelled data

Typically expensive



Elongated
Round
Unlabelled



ScaDS.AI DRESDEN LEIPZIG
Robert Haase
@haesleinhuepf
AI4Seismology
May 5th 2025

TECHNISCHE UNIVERSITÄT DRESDEN
UNIVERSITÄT LEIPZIG

CENTER FOR SCALABLE DATA ANALYTICS AND ARTIFICIAL INTELLIGENCE

Funded by

Bundesministerium für Bildung und Forschung

SACHSEN

Diese Maßnahme wird gefördert durch die Bundesregierung aufgrund eines Beschlusses des Deutschen Bundestages. Diese Maßnahme wird mitfinanziert durch Steuermittel auf der Grundlage des von den Abgeordneten des Sächsischen Landtags beschlossenen Haushaltes.

Unsupervised Machine Learning

Robert Haase

Reusing materials from Johannes Soltwedel, Till Korten, Johannes Müller, Laura Žiguty (TU Dresden), Ryan Savill (MPI-CBG), Matthias Täschner (ScaDS.AI/Uni Leipzig) and the Scikit-learn community.

Hypothesis-driven quantitative science

Hypothesis: The amplitude of a given signal is an indicator for upcoming earthquakes.

Null-Hypothesis: There is no relationship between the amplitude and future earthquakes.

Data download

Shall we use a different dataset / sensor?

Data preprocessing

Shall we use a different denoising algorithm?

Shall we modify our measurement + hypothesis?

Amplitude measurement

Statistics

Reject / accept null-hypothesis

Shall we use a different statistical test?

Be careful going down this rabbit hole, you may be leaving good scientific practice behind.

Data-driven quantitative science

~~Hypothesis: The amplitude of a given signal is an indicator for upcoming earthquakes.~~

Question: Which measurement is a good predictor for upcoming earthquakes?

Which sensor / data is the most reliable?

Data download (multiple sources, sensors, ...)

Why?

Data preprocessing using Method A, B, C

Amplitude, frequency, wavelength, ... measurement

Which parameter shows any relationship with upcoming earthquakes?

Statistics

Hypothesis generation

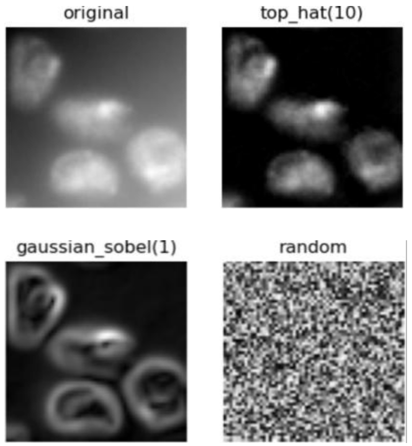
Feature selection

- Which measurement / parameter / feature is related to the effect I'm investigating?
- Example goals:

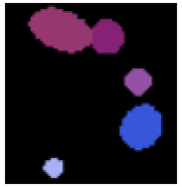
- Amplitude
- Energy
- Duration
- …

- Silence
- Tourists jumping on a sensor
- Earthquake approaching

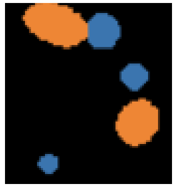
Signal classification



Pixel classification



- Area
- Perimeter
- Aspect ratio
- …



- Round
- Elongated

Object classification

Feature selection

Question: Which features shall I analyse?

Challenges:

- Physical properties versus measurable features
- Correlation versus causation
- Too many features

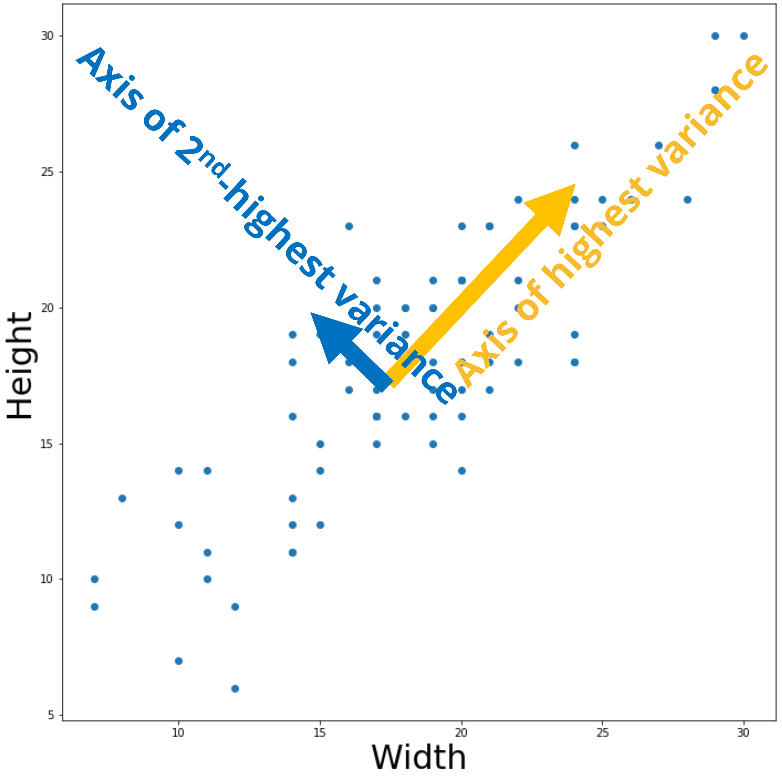
If you have no idea -> unsupervised machine learning

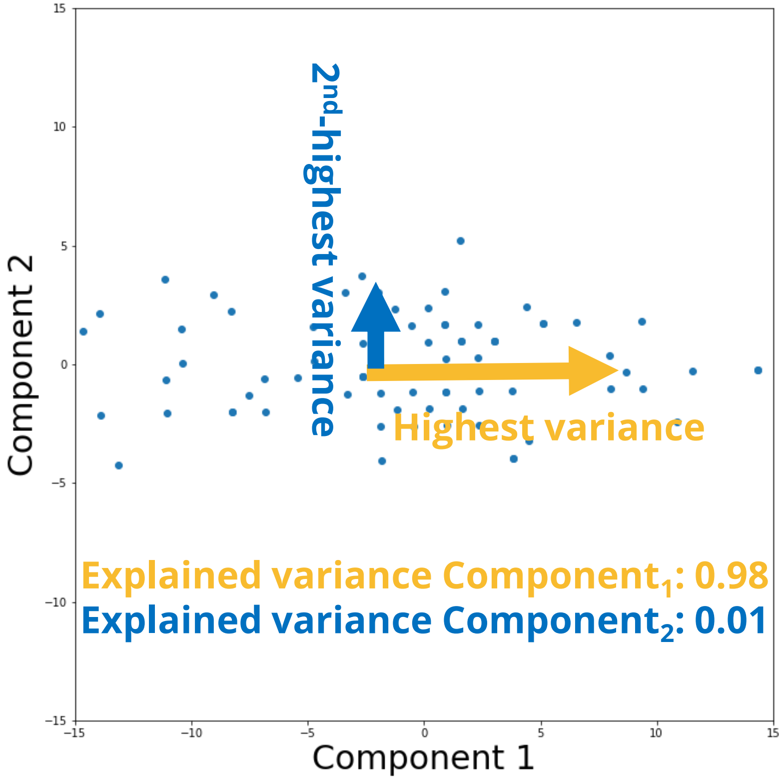
- Dimensionality reduction
- Clustering

Dimensionality reduction: Principal Component Analysis (PCA)

Linear transformation of high-dimensional data to concentrate information in a lower dimensional *embedding*

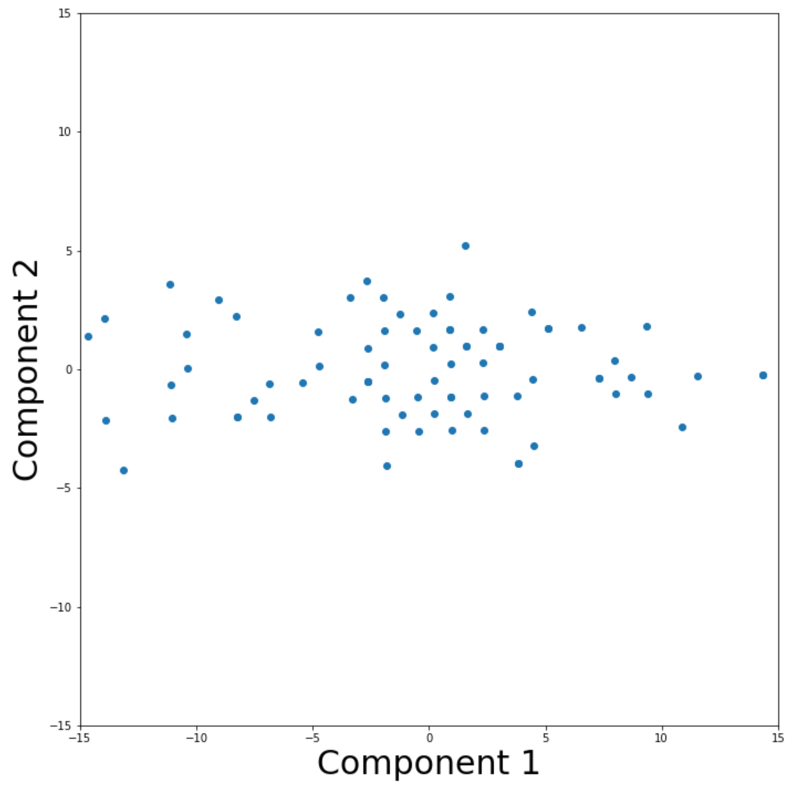
	height	width	depth
0	0.649060	0.213074	0.032167
1	0.983763	0.533933	0.026125
2	0.826448	0.223712	0.048805
3	0.610540	0.574425	0.116101
4	0.383580	0.042504	0.973645
5	0.222935	0.842952	0.152771
6	0.946367	0.780378	0.565486
7	0.580490	0.001958	0.945884
8	0.005322	0.019889	0.455281
9	0.359661	0.426161	0.369291





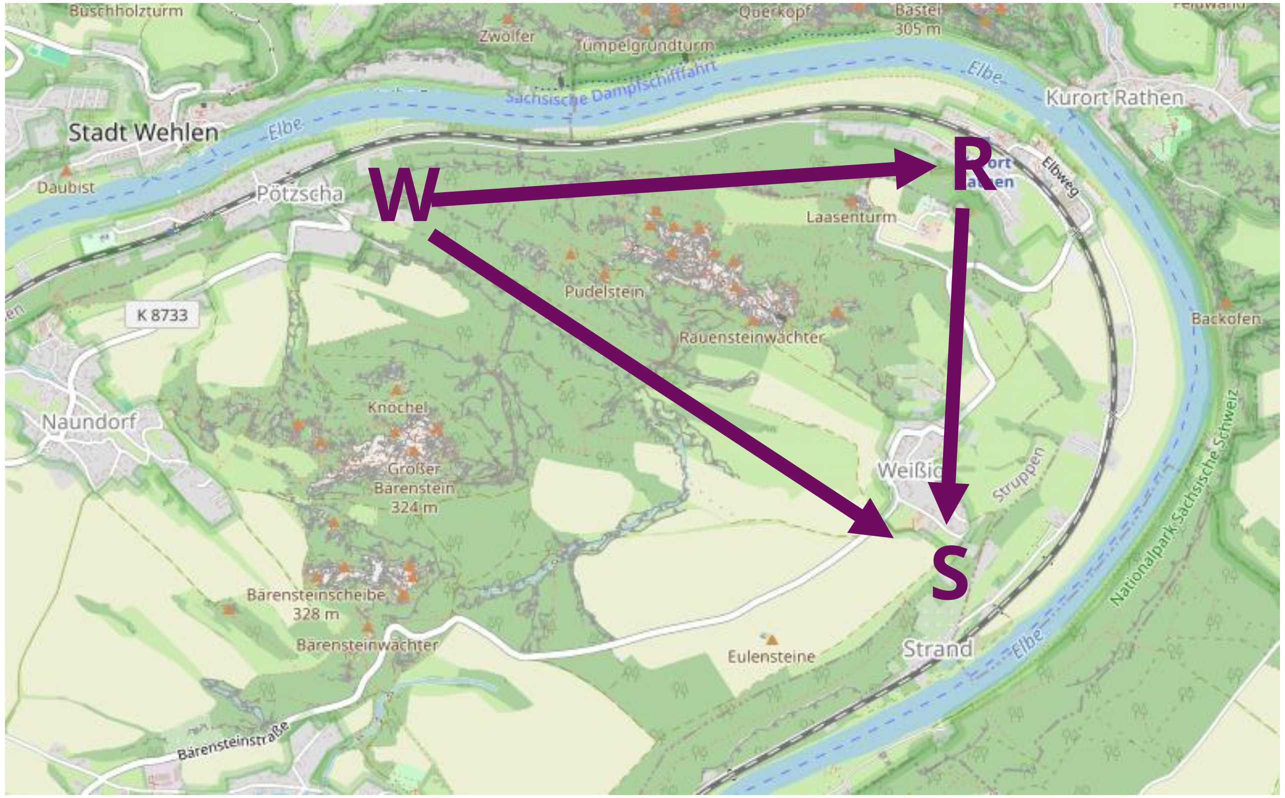
Embeddings

- N-dimensional latent space
- Axes typically have no meaningful/physical name (PCA1, UMAP1, …) and no physical unit
- Allow representing complex measurements, things, relationships in numeric space.
- Example:
  - You measure amplitude, frequency, wavelength, etc.,
  - derive a 2D-embedding from it,
  - to visualize the data or
  - to better process data



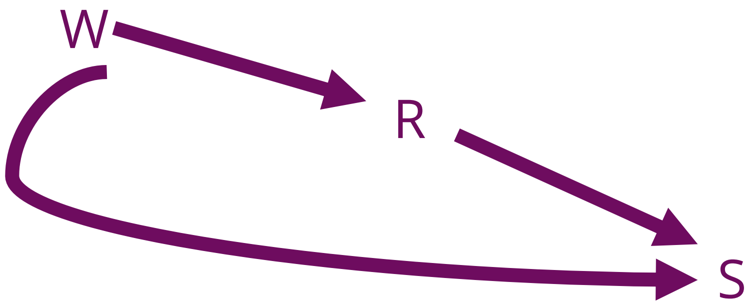
Non-Euclidian spaces

Not all features might be distances



Use travel time between W and S as metric for distance

→ Travelling from **W**ehlen to **S**trand by bike is probably faster if you make a detour through **R**athen



Uniform Manifold Approximation Projection (UMAP)

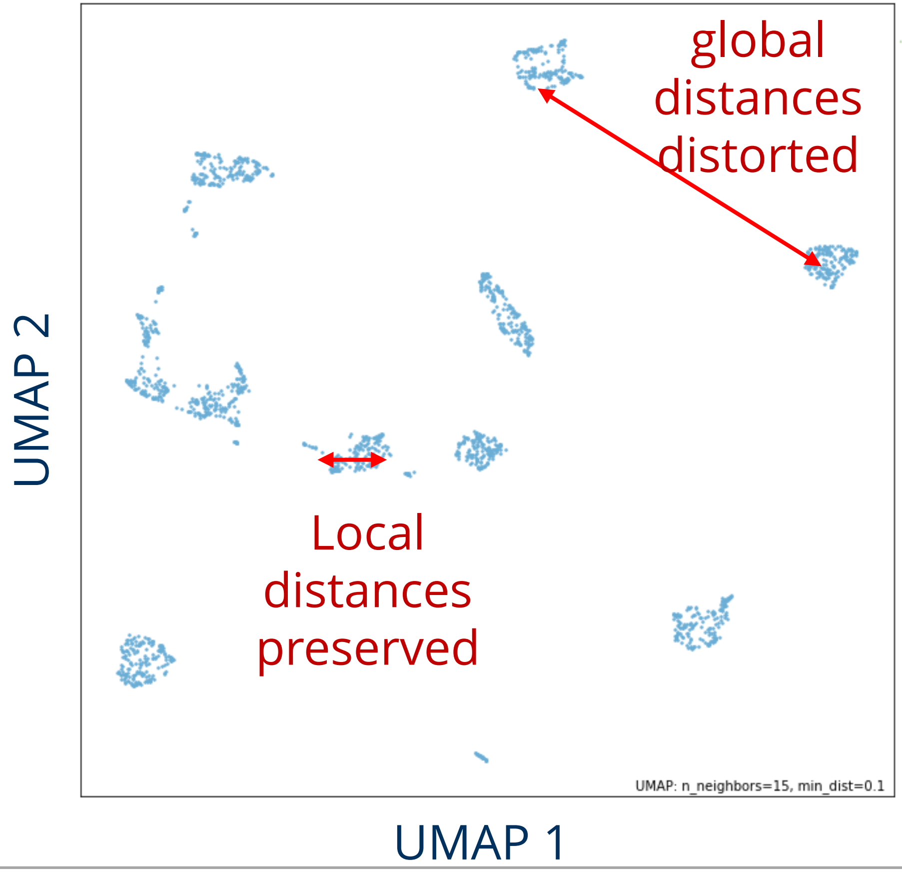
Structural, hierarchical, non-linear transformation

Modifies density of data points.

	count	mean	std
label	44.0	22.500000	12.845233
area	44.0	401.863636	202.852288
bbox_area	44.0	542.750000	295.106376
equivalent_diameter	44.0	21.781085	6.174086
convex_area	44.0	423.295455	216.613747
max_intensity	44.0	234.909091	17.517856
mean_intensity	44.0	190.116971	15.034153
min_intensity	44.0	128.000000	0.000000
extent	44.0	0.758804	0.063276
local_centroid-0	44.0	11.439824	4.126230
local_centroid-1	44.0	10.138666	3.491815
solidity	44.0	0.953153	0.024749
feret_diameter_max	44.0	26.382434	8.915046
major_axis_length	44.0	25.876797	9.591558
minor_axis_length	44.0	18.872898	5.158791
orientation	44.0	0.053057	0.691430
eccentricity	44.0	0.600434	0.165688
standard_deviation_intensity	44.0	29.556705	5.507399
aspect_ratio	44.0	1.374342	0.397611
roundness	44.0	0.762889	0.156695
circularity	44.0	0.918858	0.133288

Many dimensions

UMAP



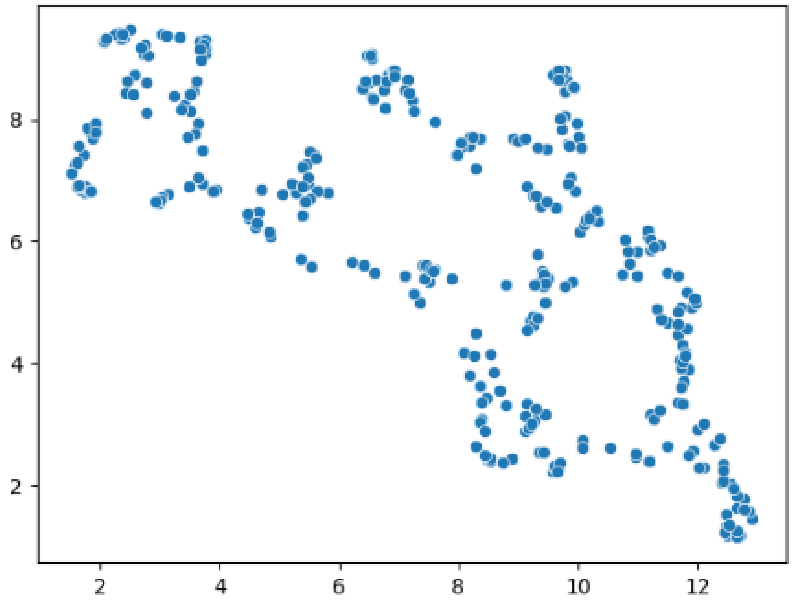
Uniform Manifold Approximation Projection (UMAP)

Non-deterministic algorithm: You execute it twice, you get different results.

[11]:
```
reducer = umap.UMAP()
embedding2 = reducer.fit_transform(scaled_statistics)

seaborn.scatterplot(x=embedding2[:, 0],
                    y=embedding2[:, 1])
```

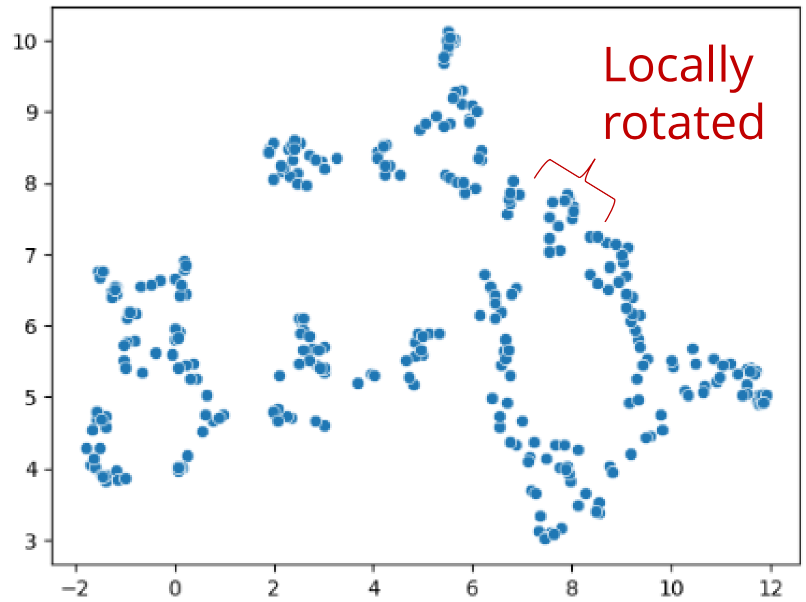
[11]: <AxesSubplot: >



[12]:
```
reducer = umap.UMAP()
embedding2 = reducer.fit_transform(scaled_statistics)

seaborn.scatterplot(x=embedding2[:, 0],
                    y=embedding2[:, 1])
```

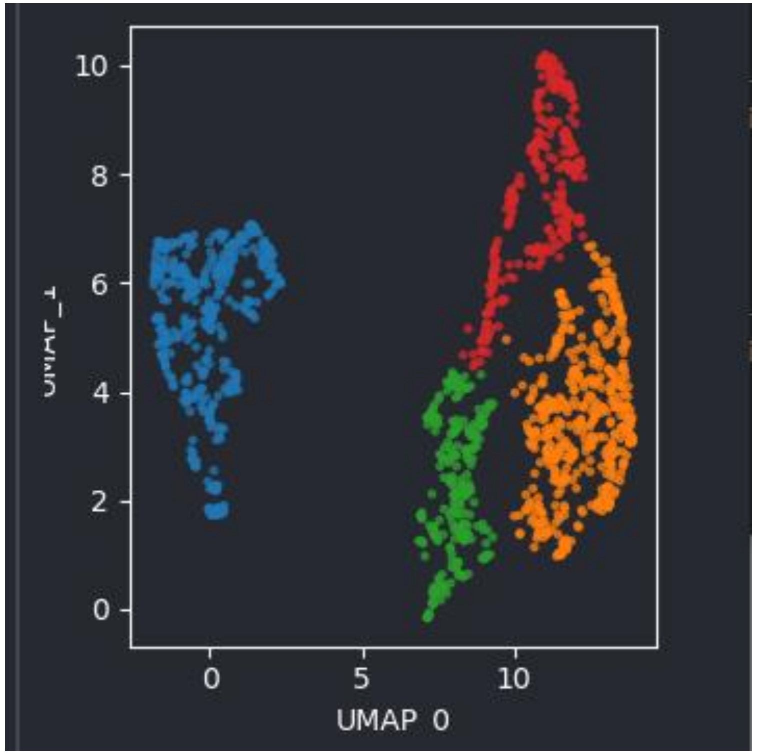
[12]: <AxesSubplot: >



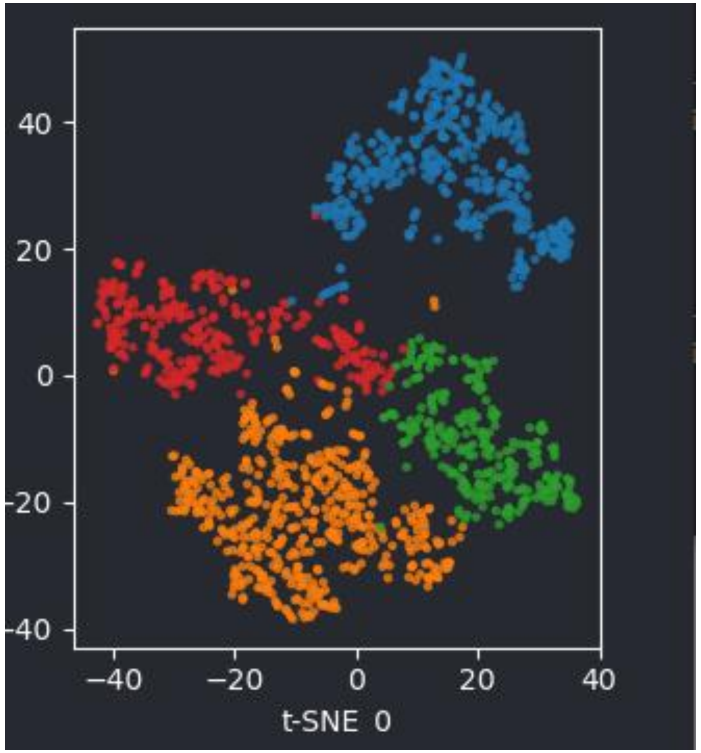
Globally rotated

Dimensionality reduction

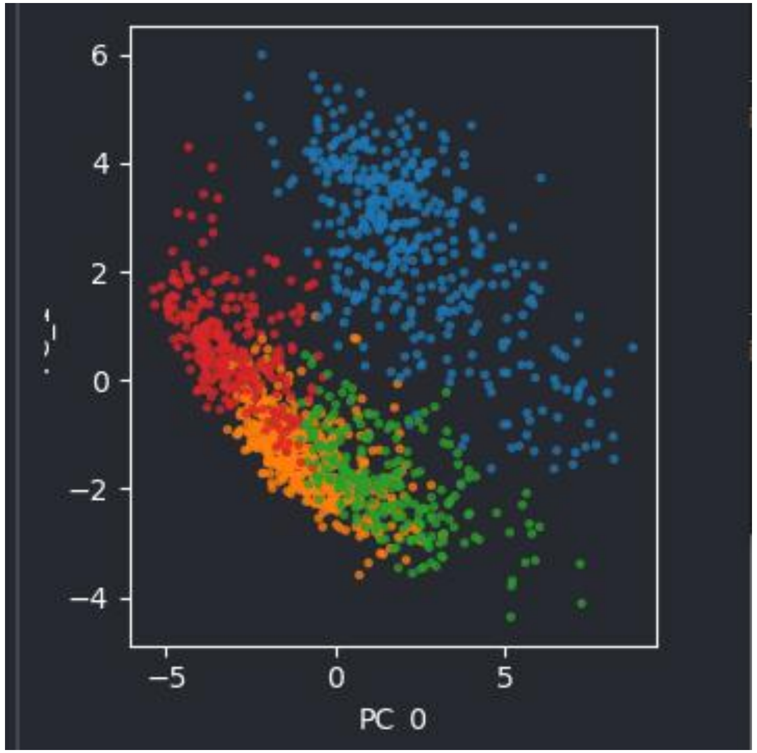
Uniform manifold approximation and projection (UMAP)



t-distributed stochastic neighbor embedding (t-SNE)

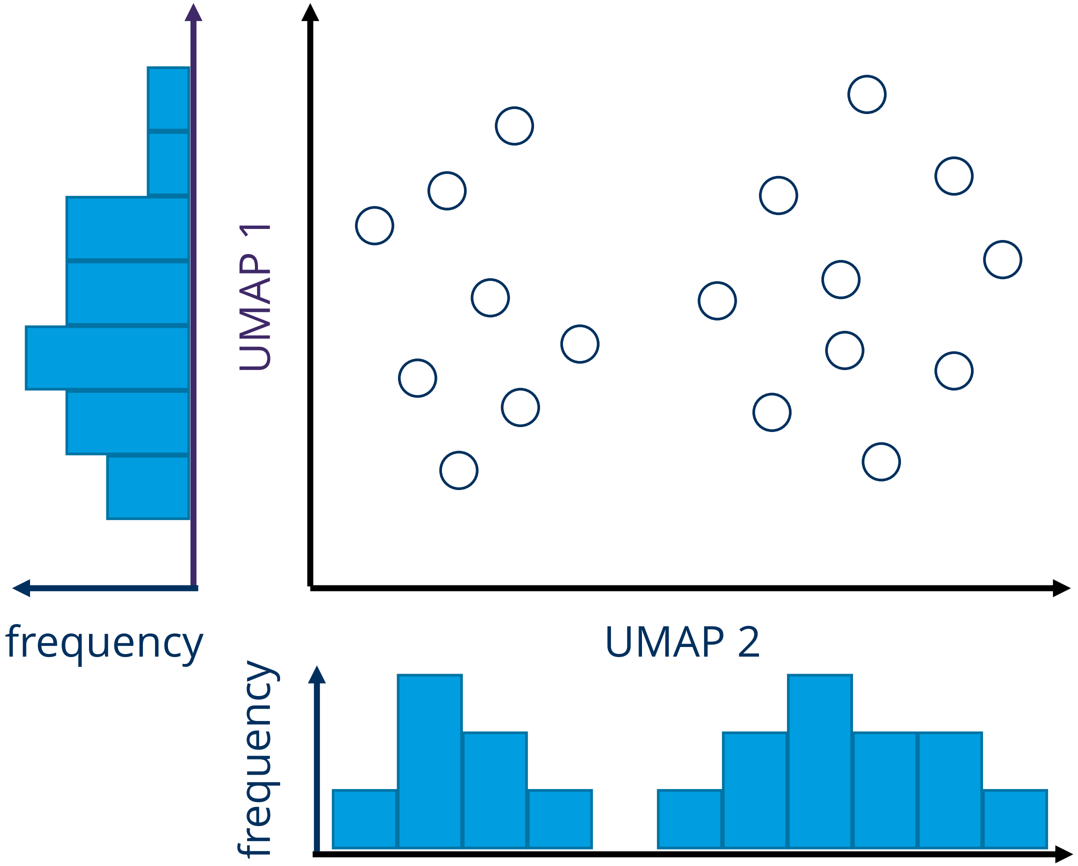


Principal component analysis (PCA)



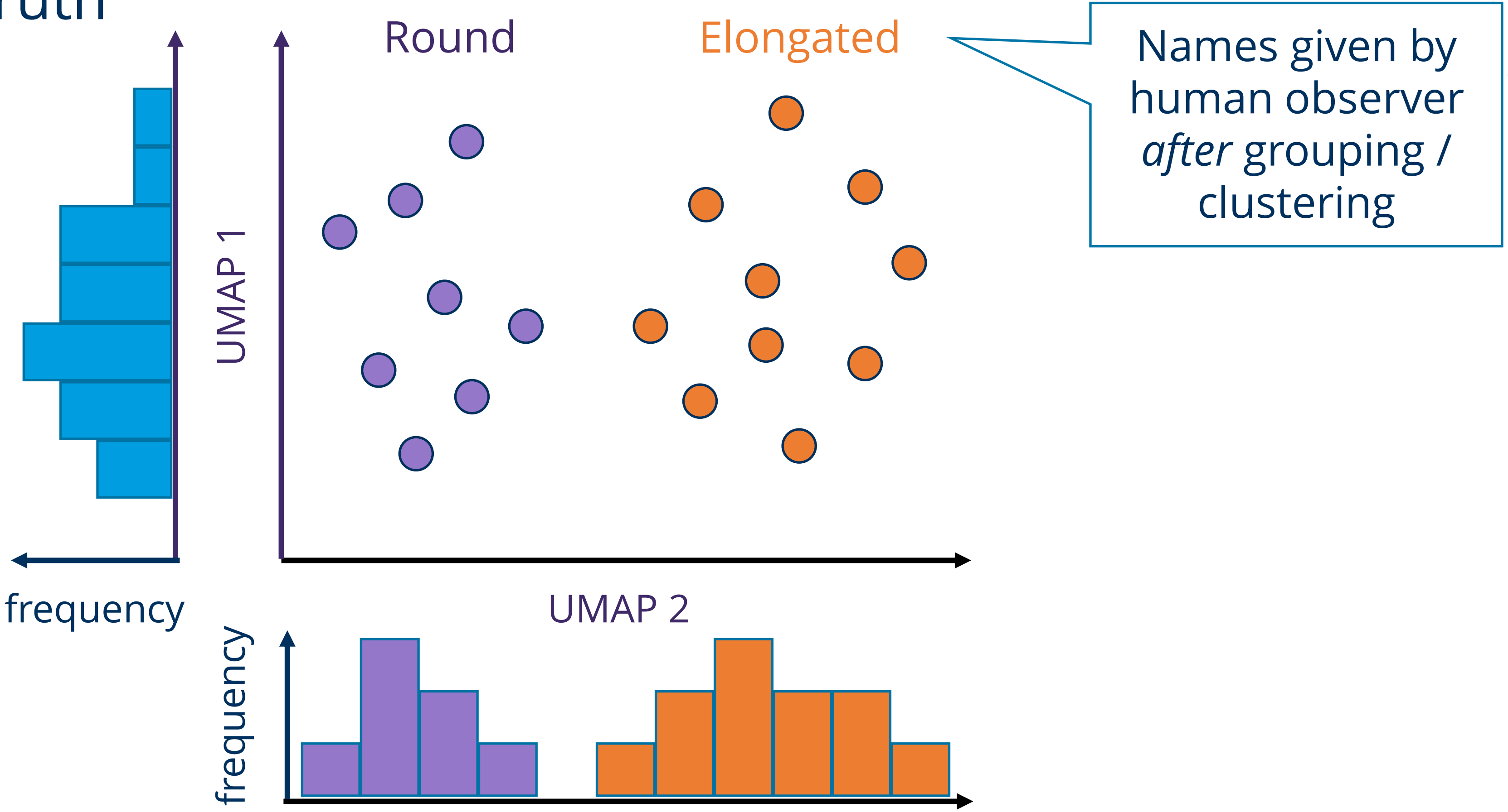
Clustering

Unsupervised machine learning may include grouping objects without given ground truth



Clustering

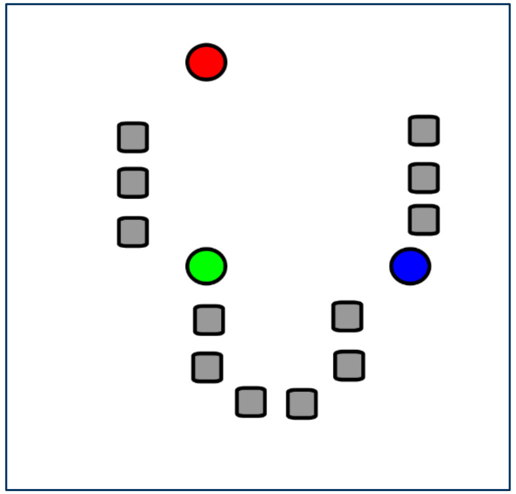
Unsupervised machine learning may include grouping objects without given ground truth



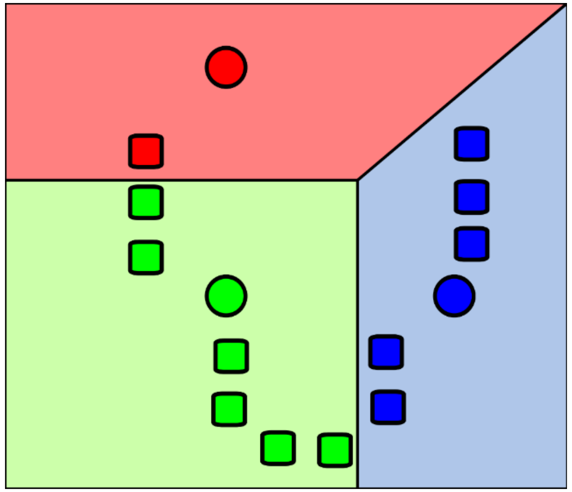
K-Means Clustering

Clustering algorithm, where you *only* need to specify the number of clusters.

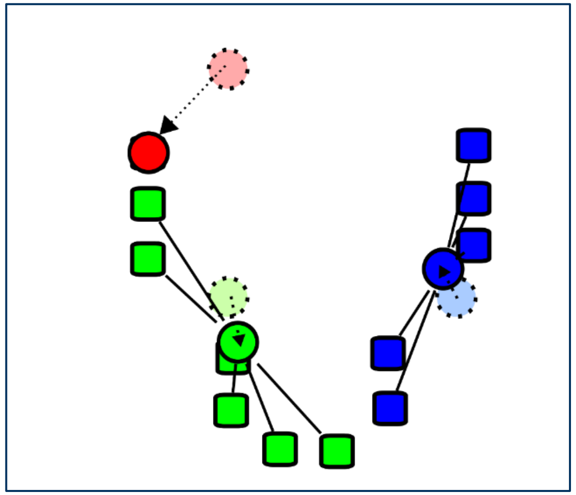
Step1: Random initialization of cluster centers



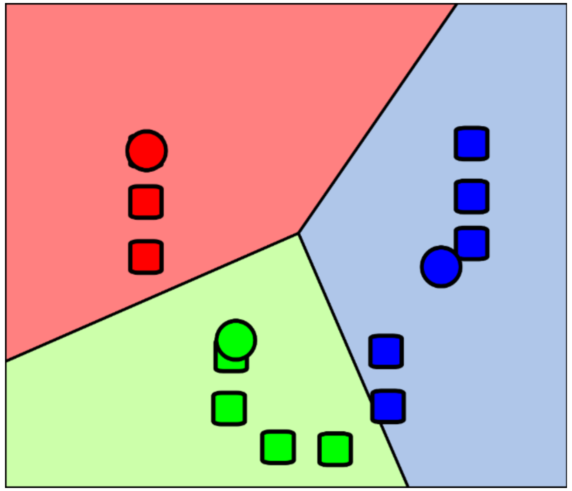
Step2: Tessellation of space into cluster regions



Step3: Replace cluster center with centroids

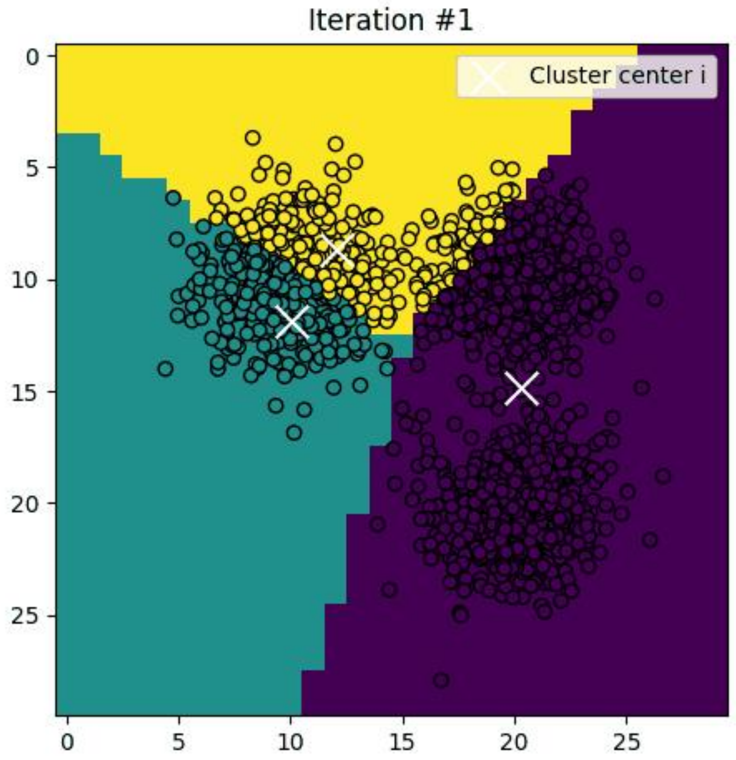


Step4: Repeat 2&3 until convergence



K-Means Clustering

Clustering algorithm, where you *only* need to specify the number of clusters.

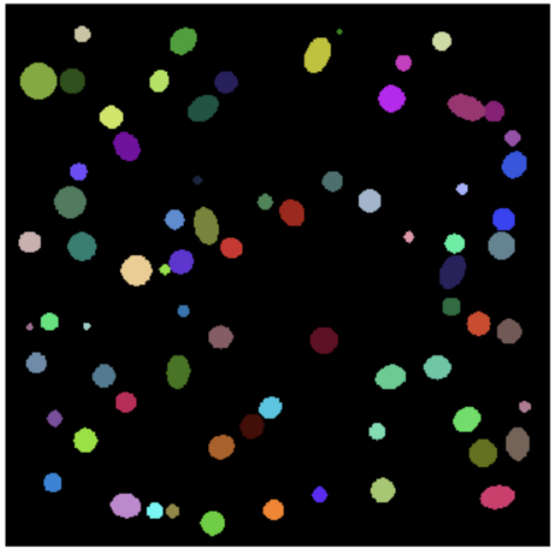


Source: I, Weston.pace, Shared under [CC-BY-SA 3.0](https://creativecommons.org/licenses/by-sa/3.0/)
https://de.wikipedia.org/wiki/K-Means-Algorithmus

Walk-through: Data Exploration

Goal: Understand shape measurements

Data: Shape measurements from *randomly* shaped blobs.



	label	area	perimeter	minor_axis_length	major_axis_length	circularity	solidity	aspect_ratio	elongation
0	1	97.0	32.970563	11.092860	11.092860	1.121318	0.788288	1.000000	0.000000
1	2	285.0	60.284271	19.052651	19.052651	0.985477	0.785116	1.000000	0.000000
2	3	473.0	79.597980	21.823280	27.594586	0.938138	0.785448	1.264456	0.209146
3	4	321.0	63.112698	19.033334	21.456036	1.012701	0.786033	1.127287	0.112915
4	5	407.0	72.769553	22.155138	23.384406	0.965839	0.785586	1.055485	0.052568

…

Walk-through: Data Exploration

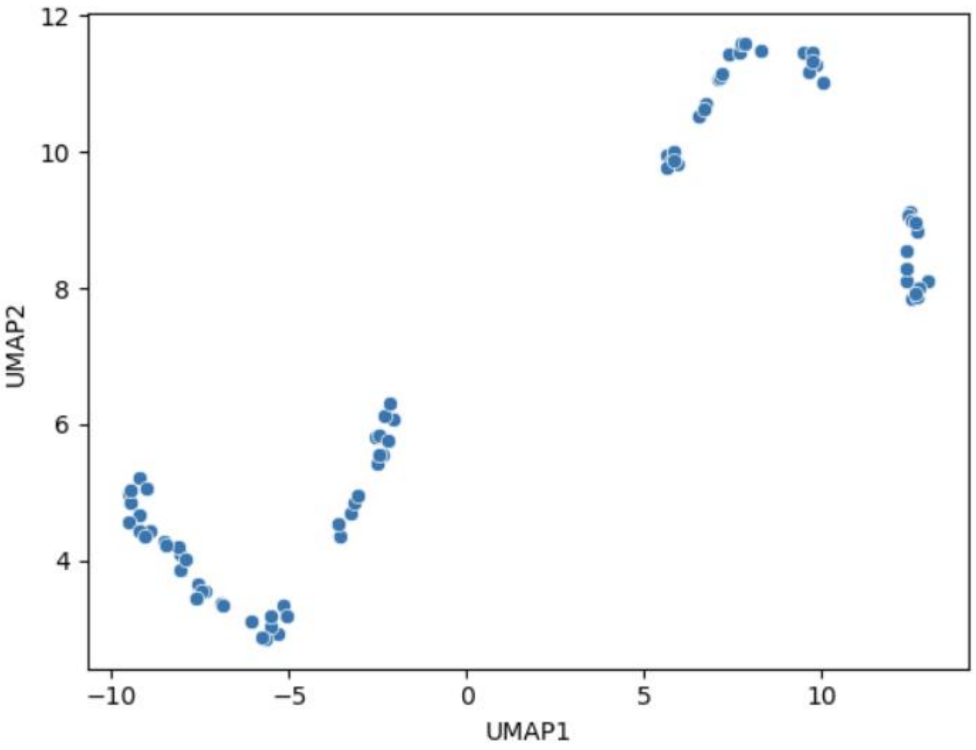
Step 1: Dimensionality reduction (UMAP)

Observation: There appear to be *2 distinct groups*

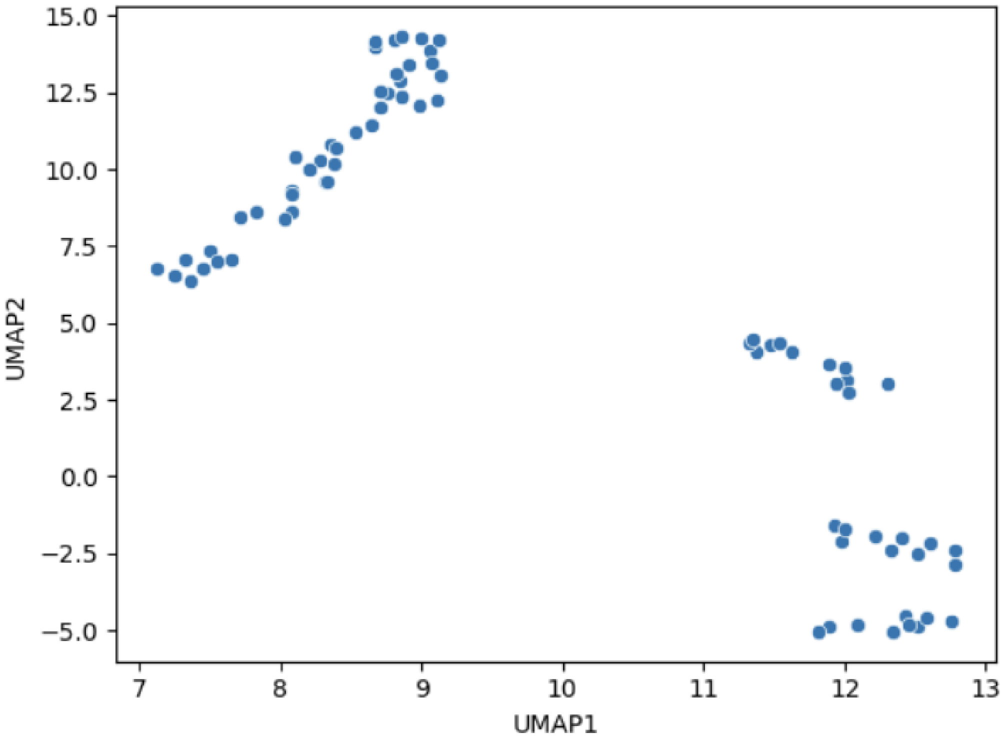
Pinning the random seed is no solution to this general problem.

Beware: UMAPs are non-deterministic. Different runs lead to different results.

Run 1:



Run 2:



Walk-through: Data Exploration

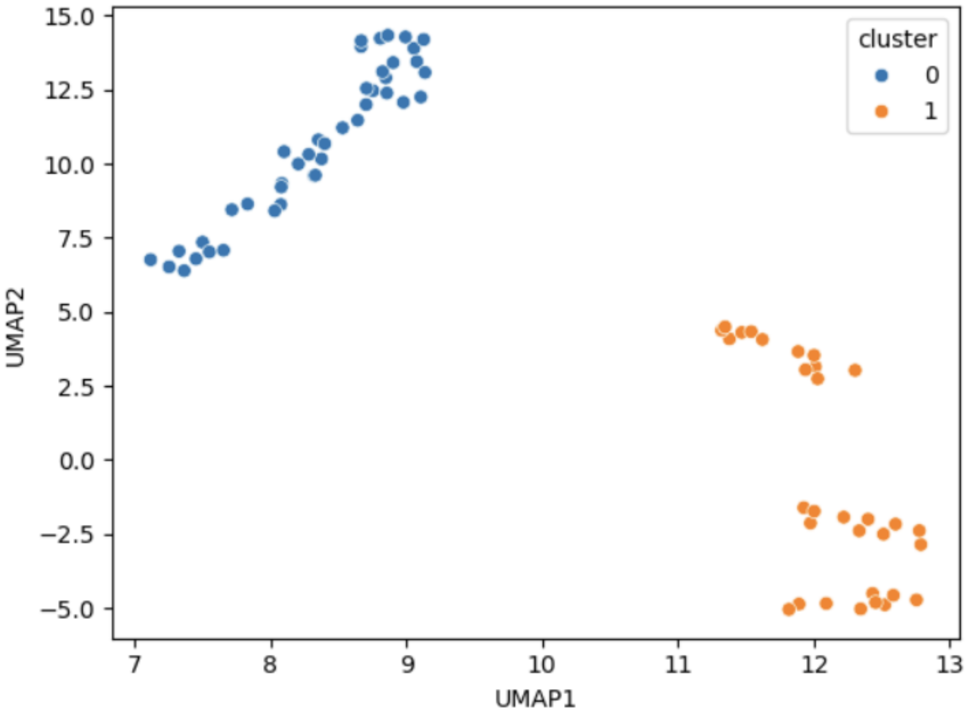
Step 2: Clustering data into 2 clusters

Using K-Means clustering

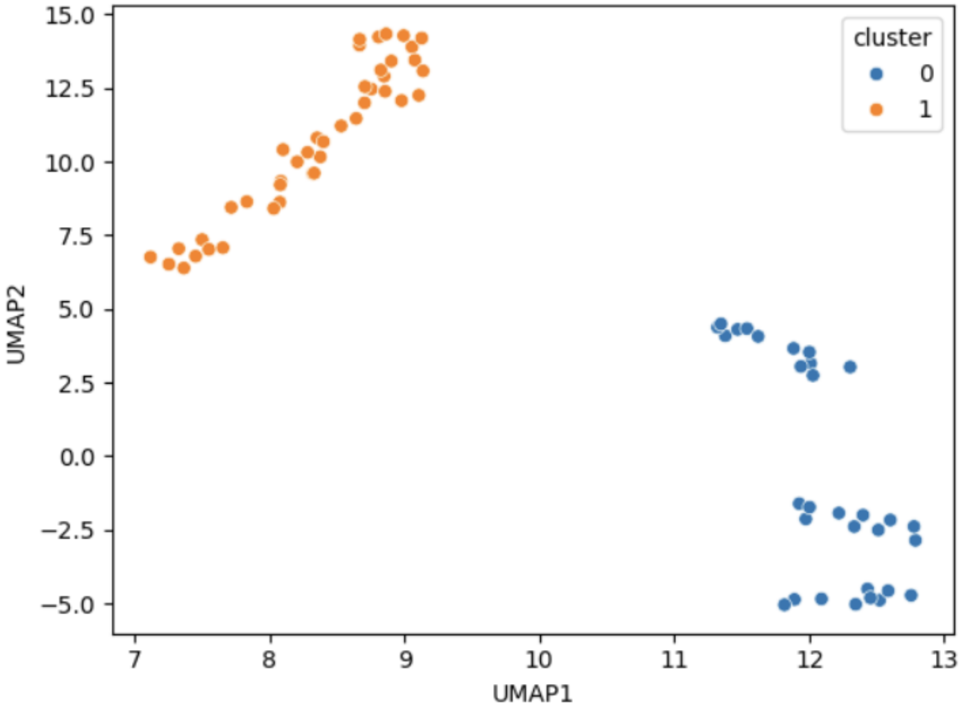
Pinning the random seed is no solution to this general problem.

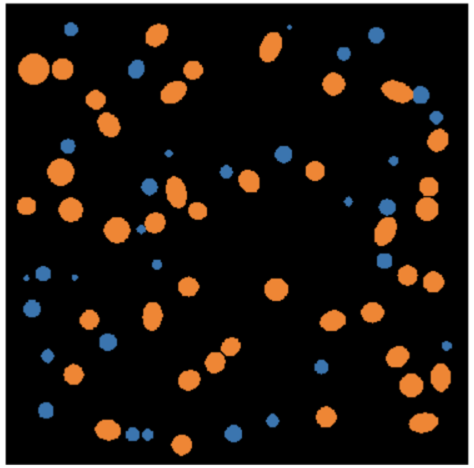
Beware: Clustering-algorithms are non-deterministic. Different runs lead to different results.

Run 1:



Run 2:



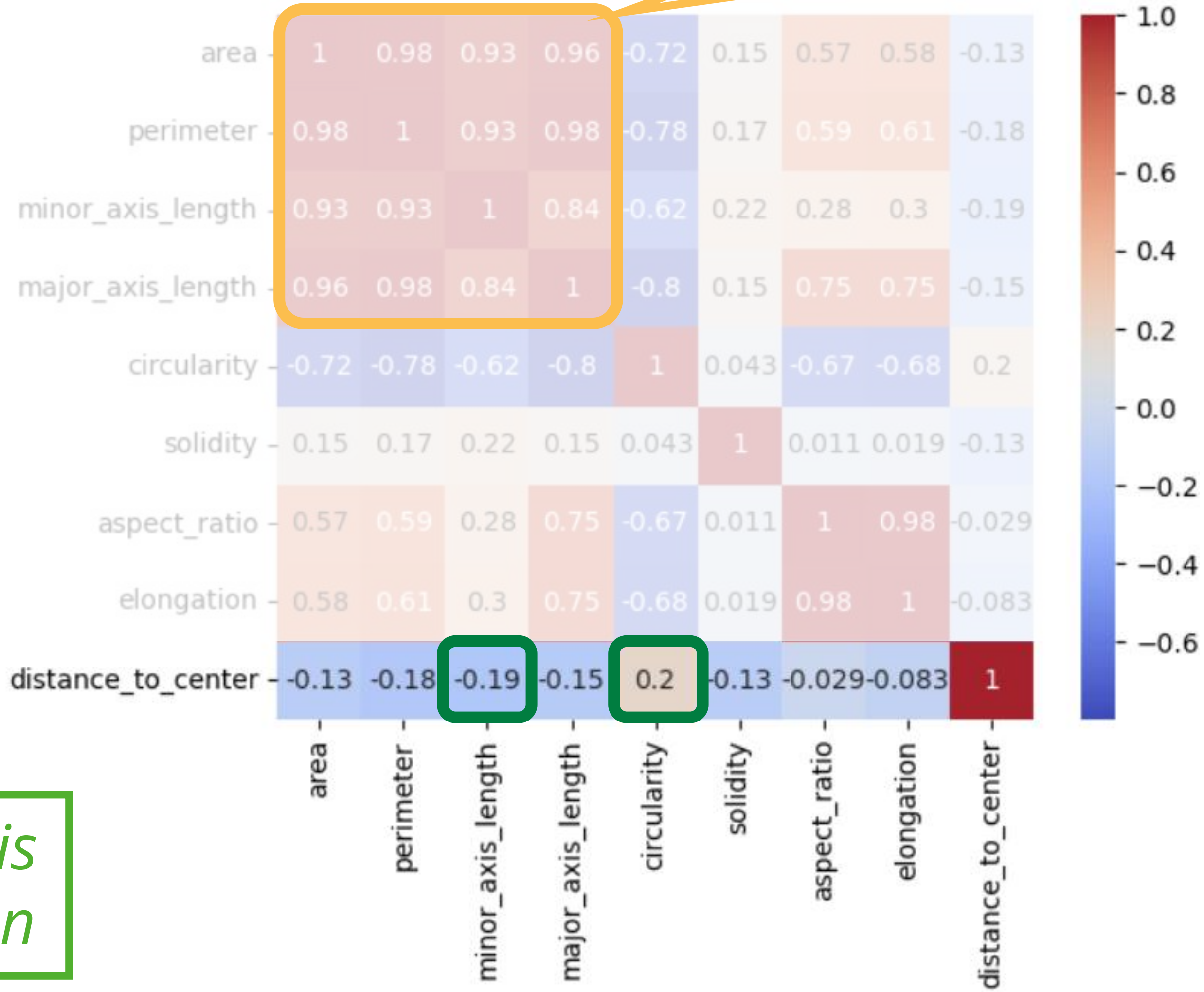


Walk-through: Data Exploration

Side note: beware of feature correlation.

Step 3: Feature selection

Based on correlation
with distance to cluster-centers



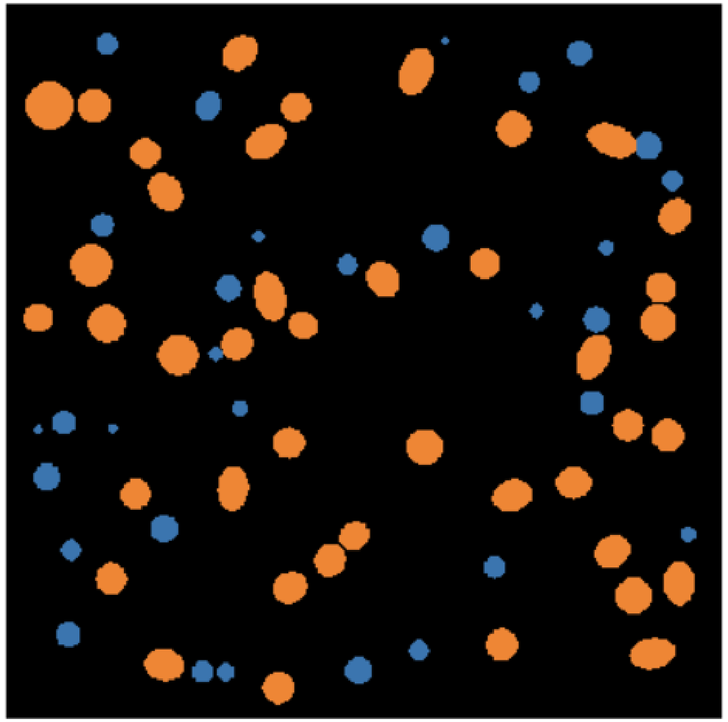
Hypothesis:
"Circularity and minor_axis_length
allow to predict classification."

Hypothesis generation

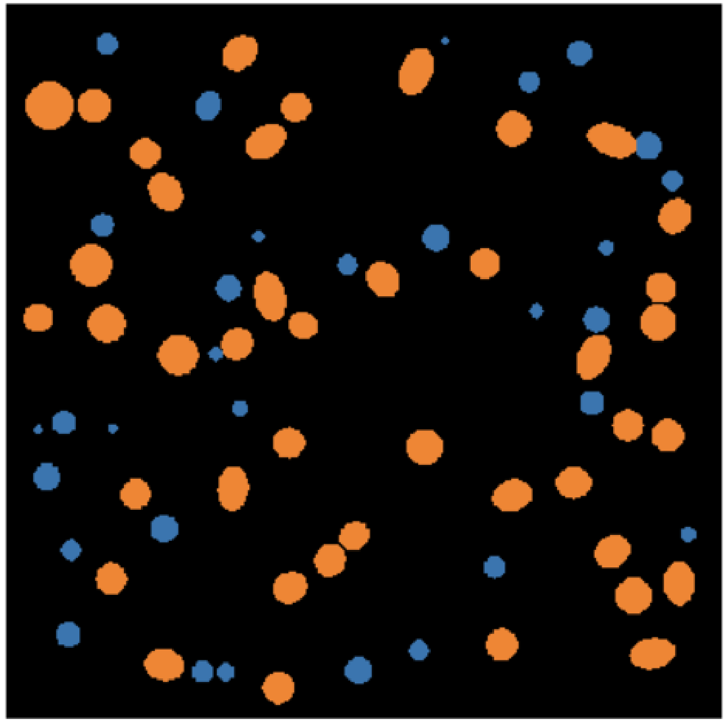
Walk-through: Data Exploration

Step 4: Train a classifier (supervised ML)

Goal: Eliminate non-determinism



Clustering result (non-deterministic)



Classification result (deterministic, repeatable)

CENTER FOR SCALABLE DATA ANALYTICS AND ARTIFICIAL INTELLIGENCE

Supervised Machine Learning

Robert Haase

Funded by

Bundesministerium für Bildung und Forschung

SACHSEN

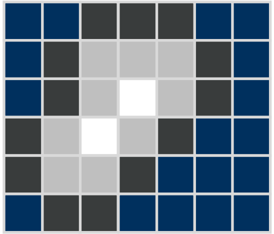
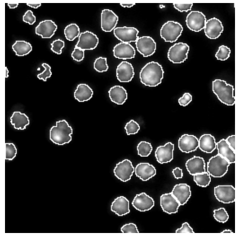
Diese Maßnahme wird gefördert durch die Bundesregierung aufgrund eines Beschlusses des Deutschen Bundestages. Diese Maßnahme wird mitfinanziert durch Steuermittel auf der Grundlage des von den Abgeordneten des Sächsischen Landtags beschlossenen Haushaltes.

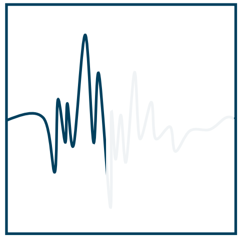
Robert Haase
@haesleinhuepf
AI4Seismology
May 5th 2025

Supervised Machine learning

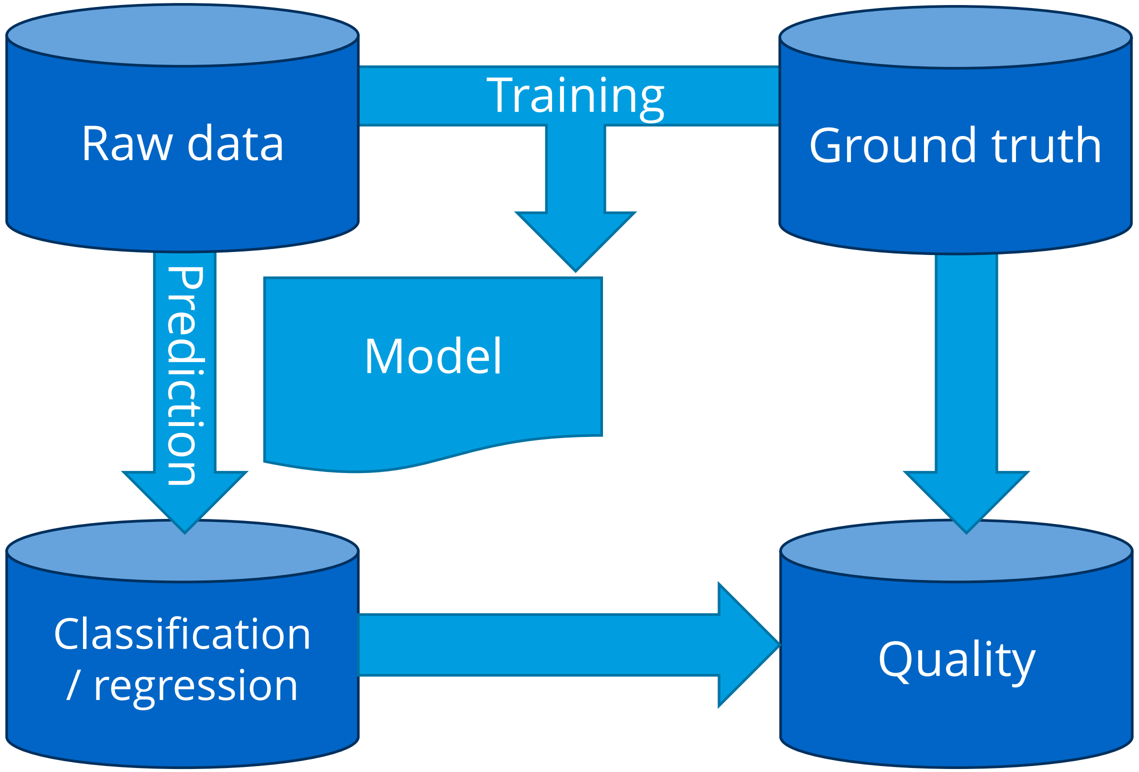
Automatic construction of predictive models from given data

Pixels, Objects, Images, Audio, Sensor data, Text, Measurements, …

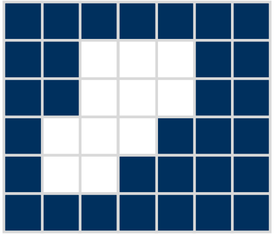
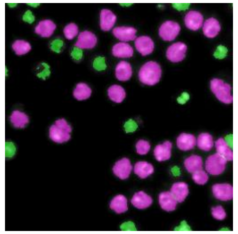





Annotated raw data, often generated by humans



Classification (categorical)

Cat
~~Dog~~

Earth-quake
~~Wind~~

Regression (continuous numerical)

n = 11

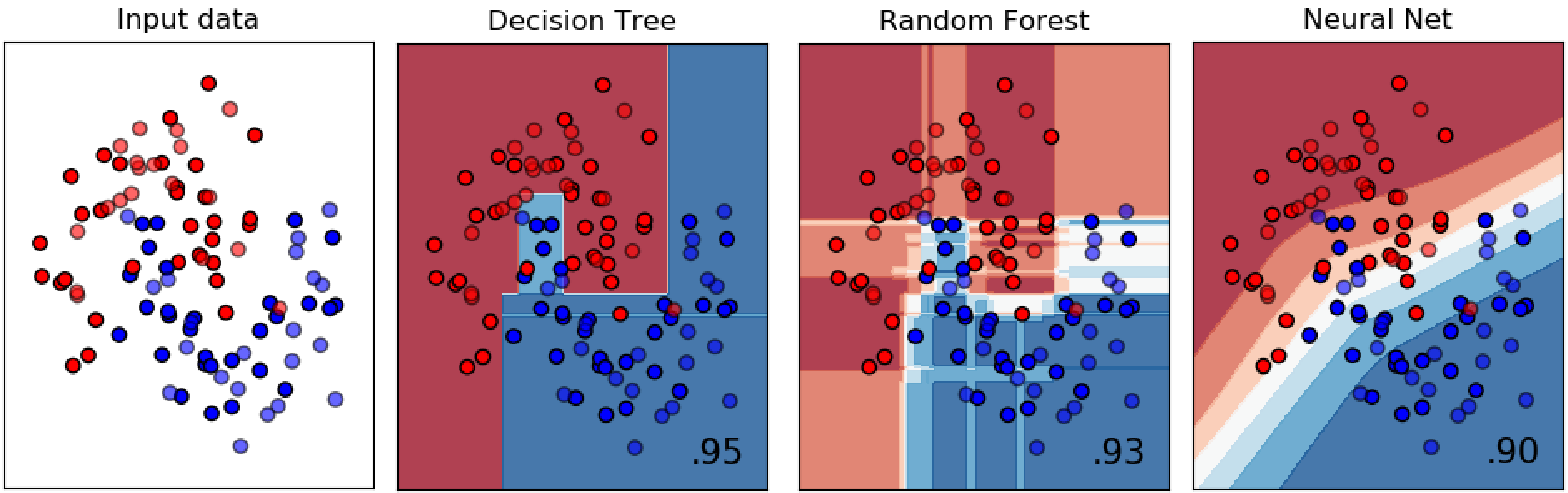
green_magenta_ratio=0.3

P_{Cat}= 0.5
$P_{Microscope}$= 0.4
Height = 80 cm

Accuracy, Precision, Recall, …

Goal

Guess classification (color) from position of a sample in parameter space.

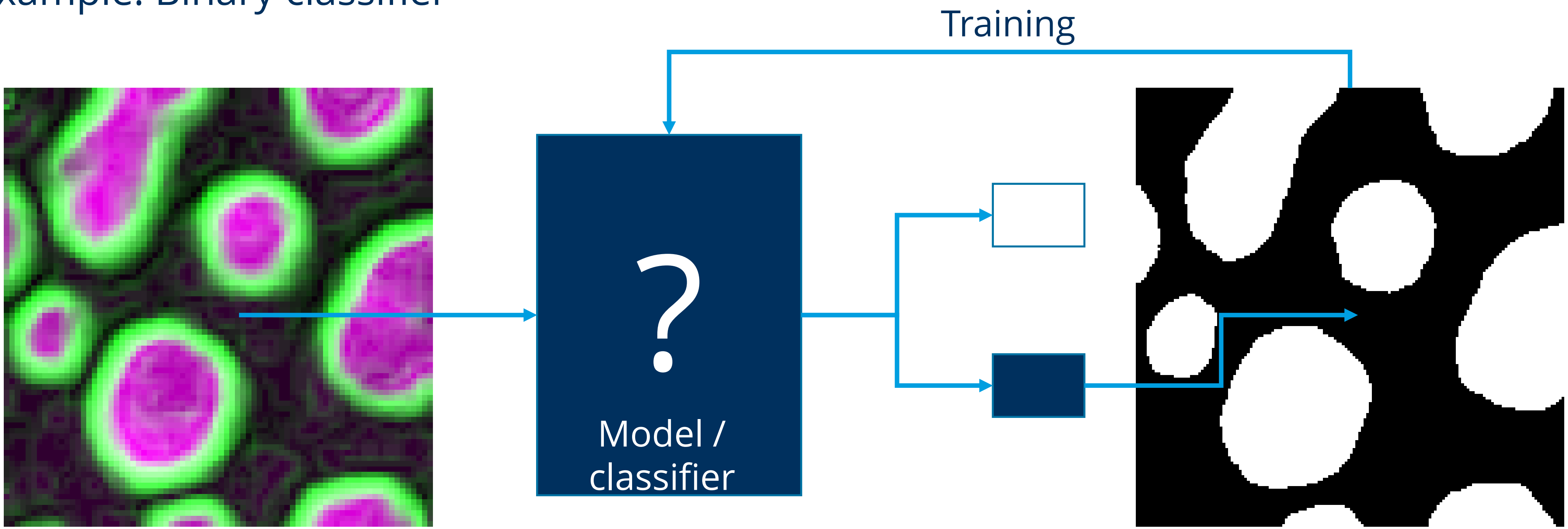


Machine learning for image segmentation

Supervised machine learning: We give the computer some ground truth to learn from

The computer derives a *model* or a *classifier* which can judge if a pixel should be foreground (white) or background (black)

Example: Binary classifier



Raw image

Binary image

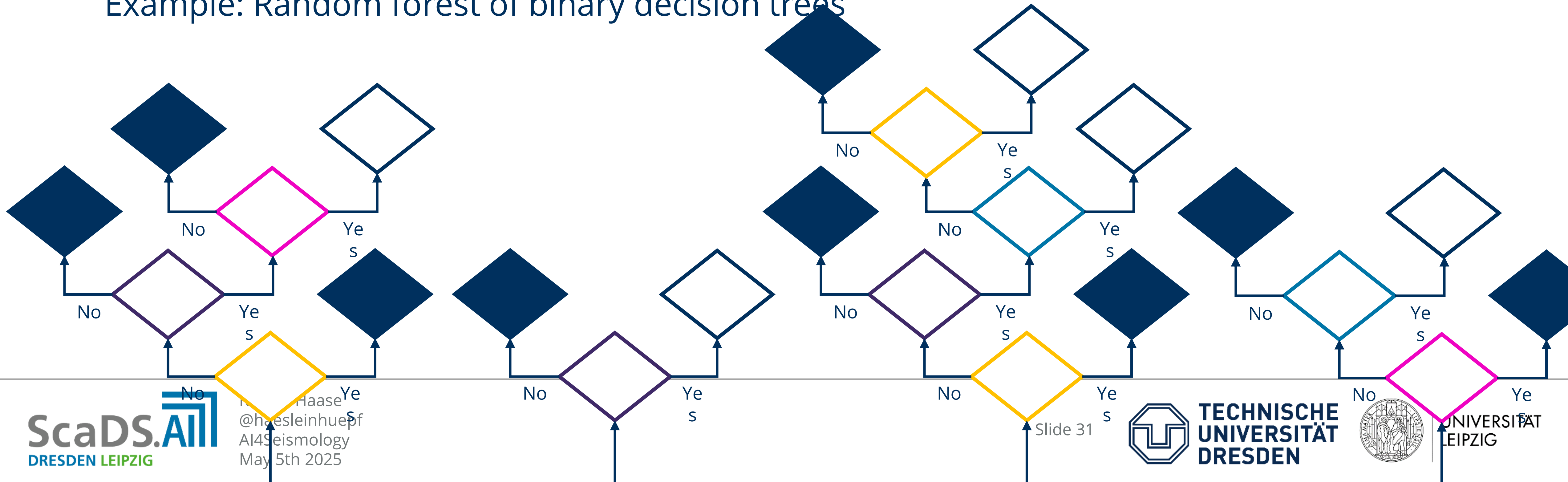
Random forest based image segmentation

Decision trees are classifiers, they decide if a pixel should be white or black

Random decision trees are randomly initialized, afterwards evaluated and selected

Random forests consist of many random decision trees

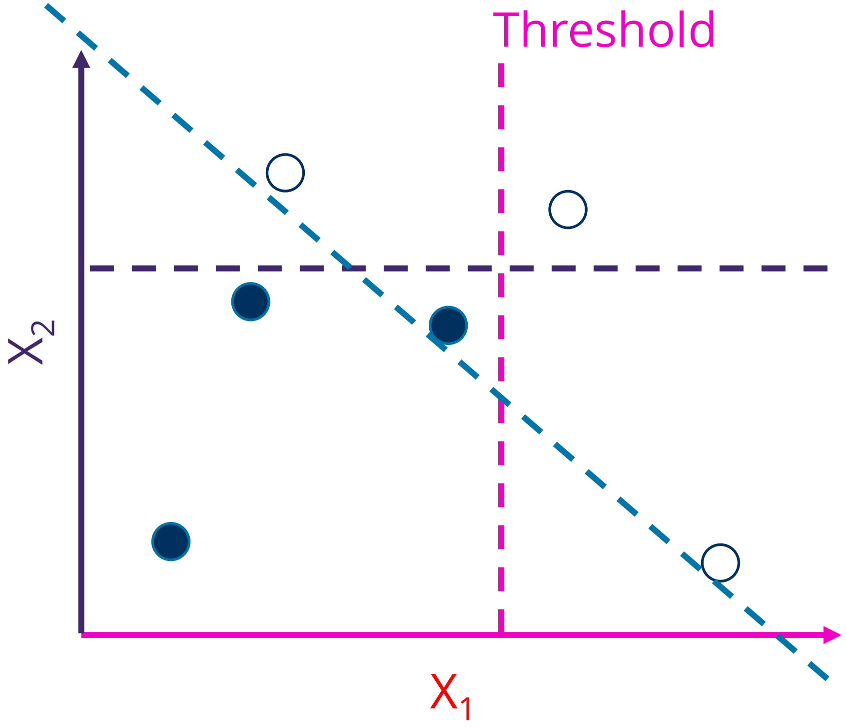
Example: Random forest of binary decision trees

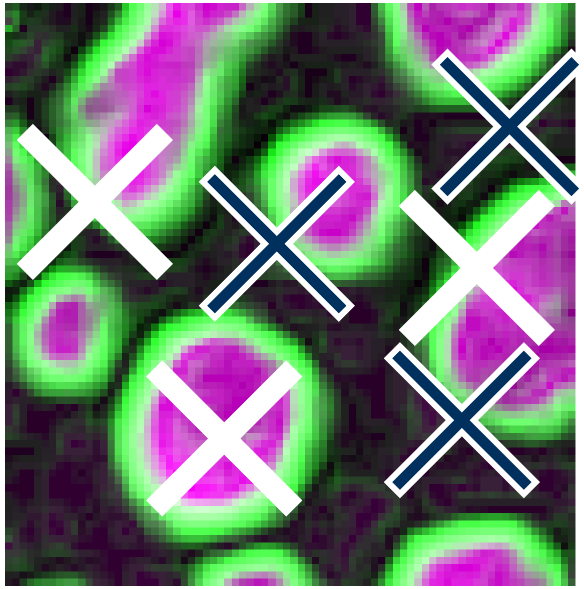


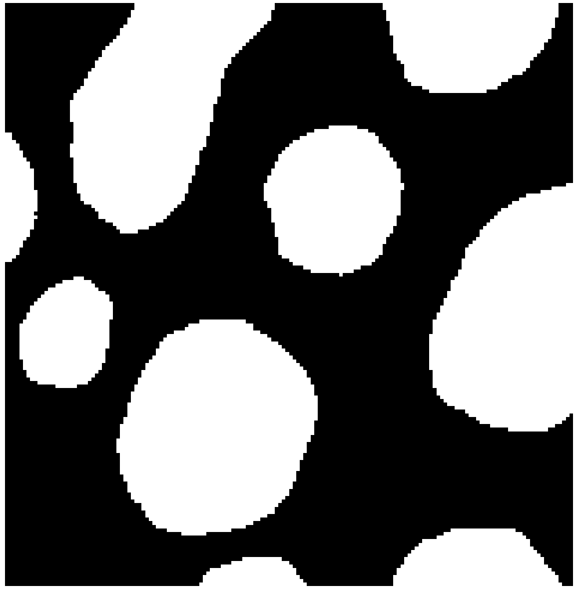
Deriving random decision trees

For efficient processing, we randomly *sample* our data set

- Individual pixels, their intensity and their classification



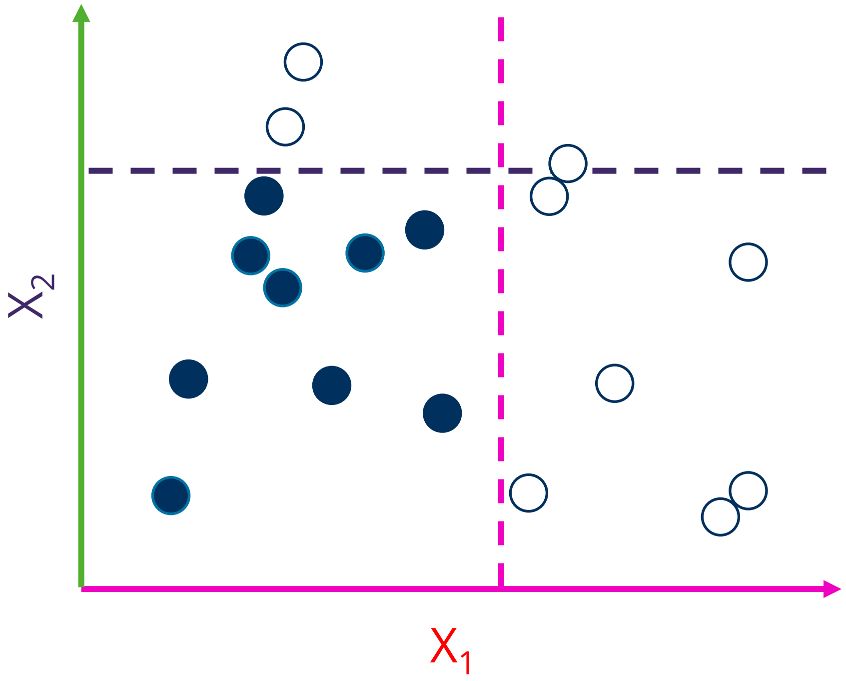


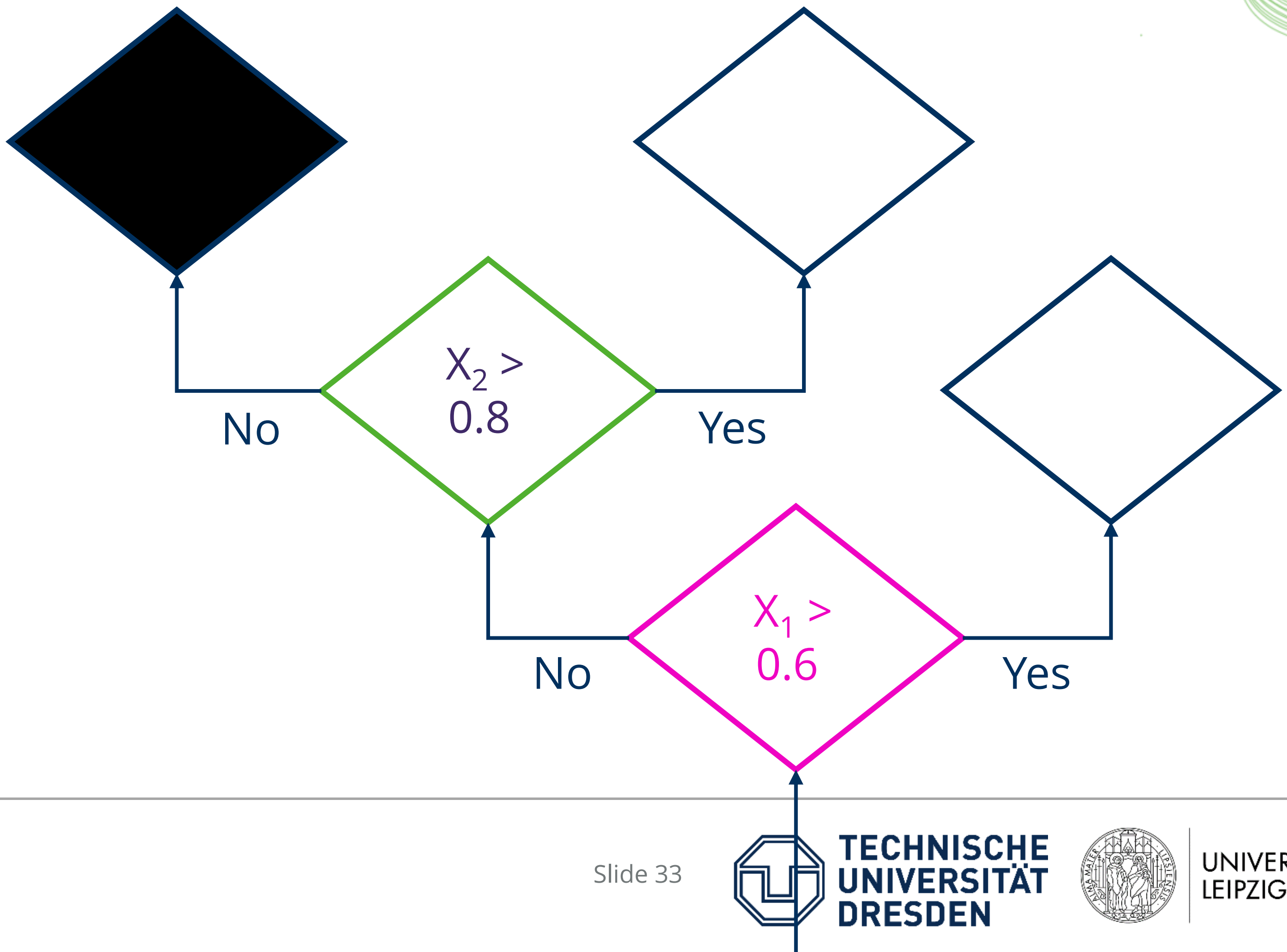


Note: You cannot use a single threshold to make the decision correctly

Deriving random decision trees

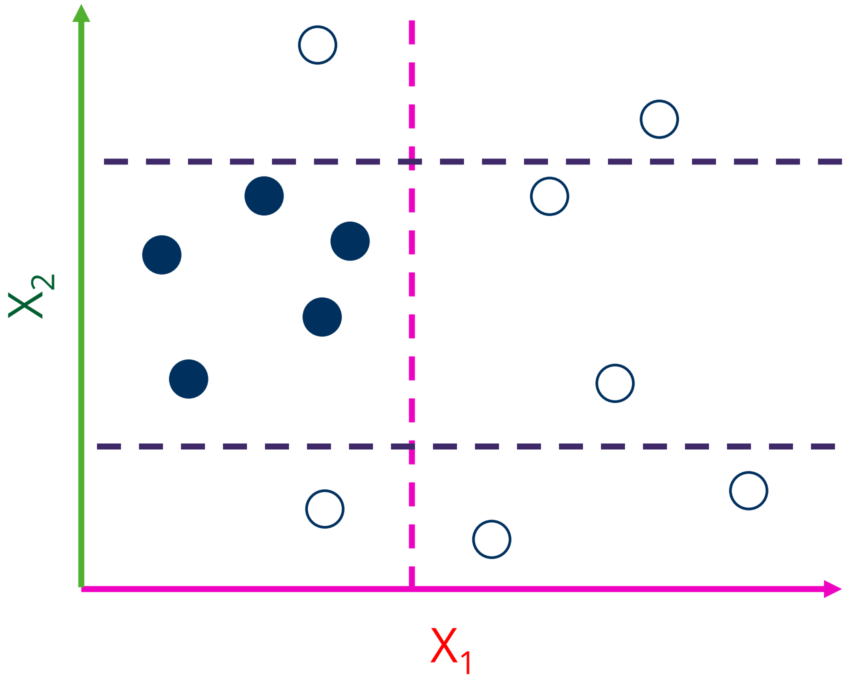
Decision trees combine several thresholds on several parameters

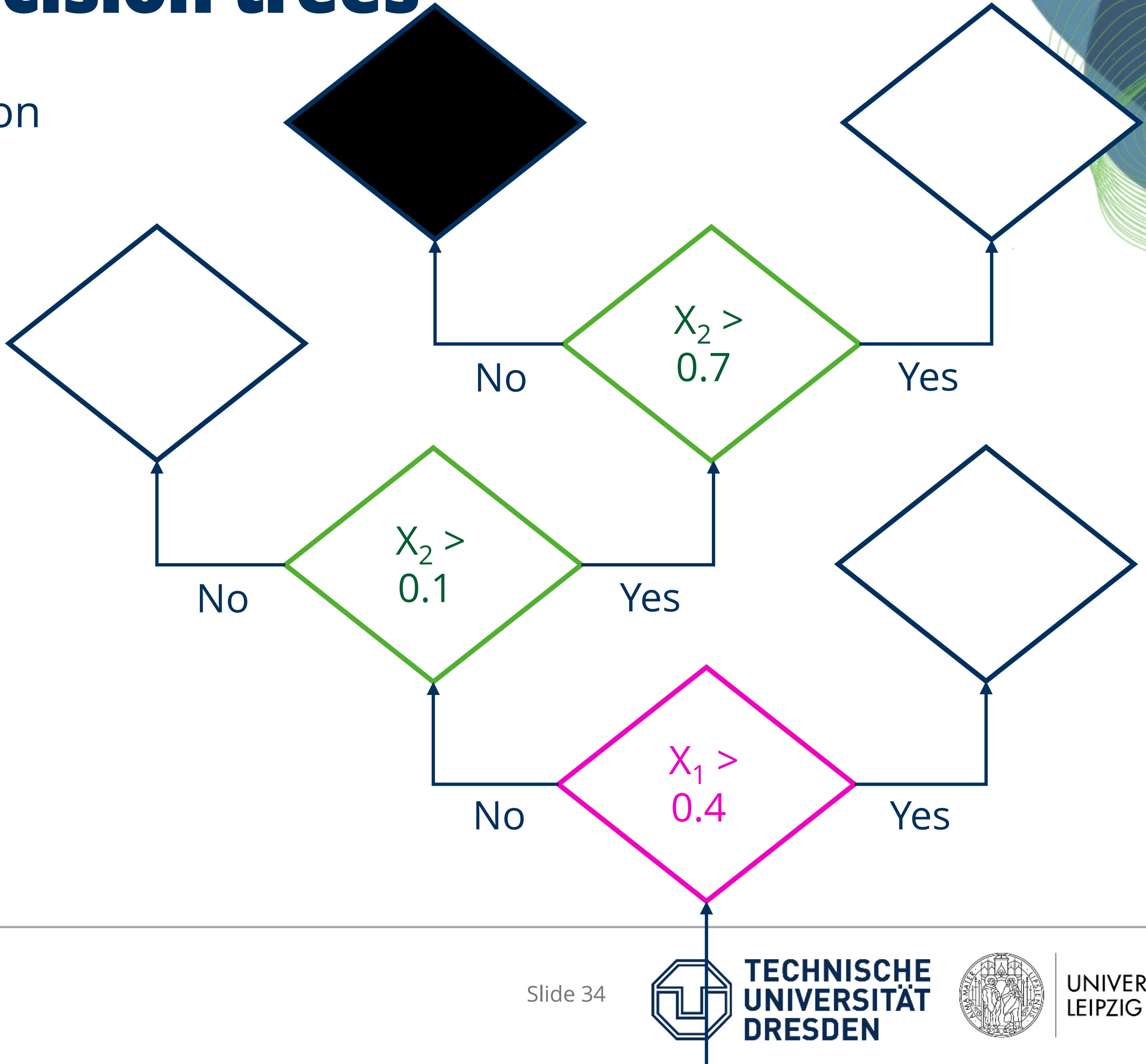




Deriving random decision trees

Depending on sampling, the decision trees are different





Random Forest Pixel Classifiers

By training many decision trees, errors are equilibrated

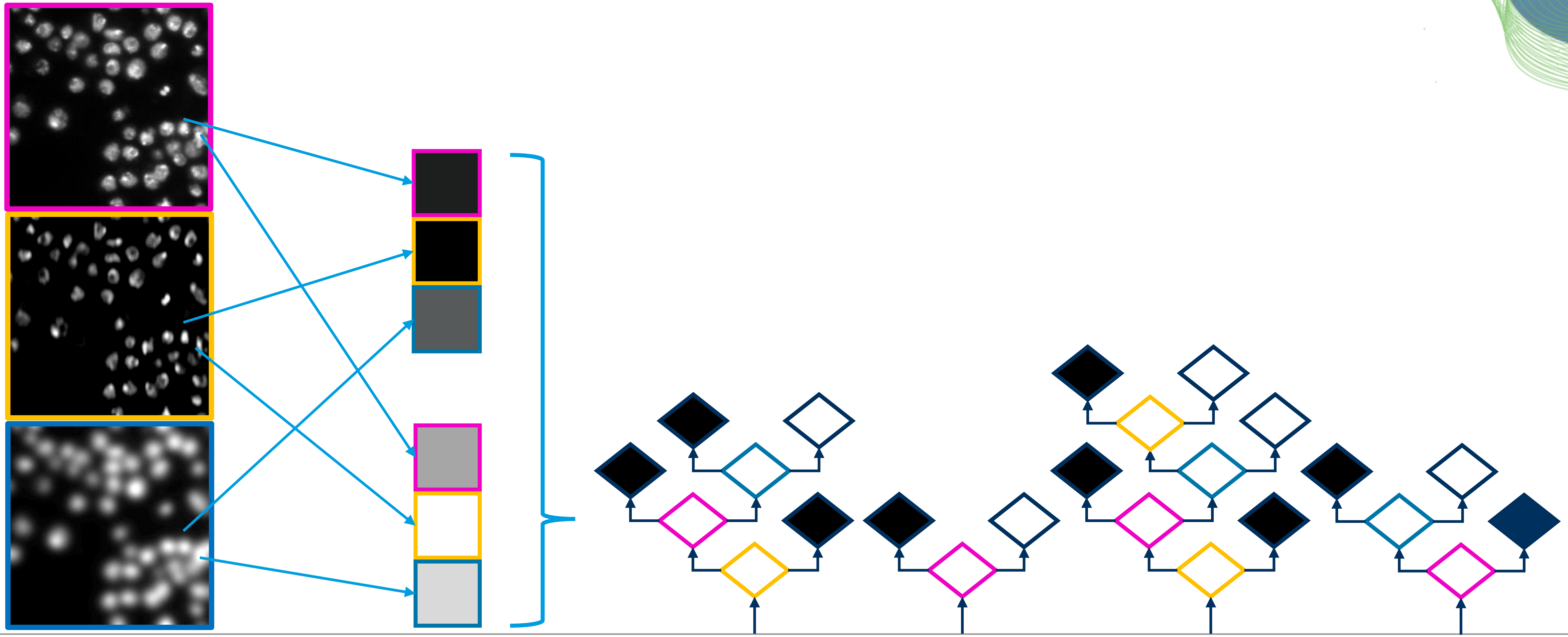
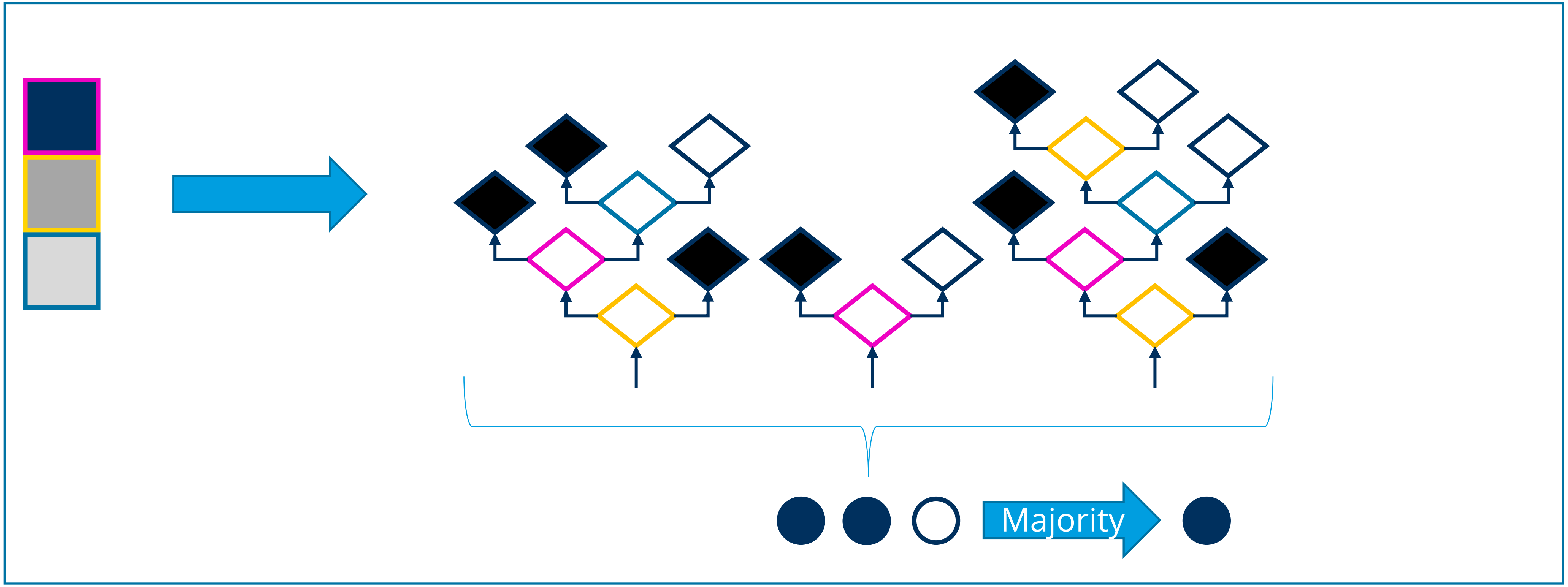


Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019)

Random Forest Pixel Classifiers

Combination of individual tree decisions by voting or max / mean

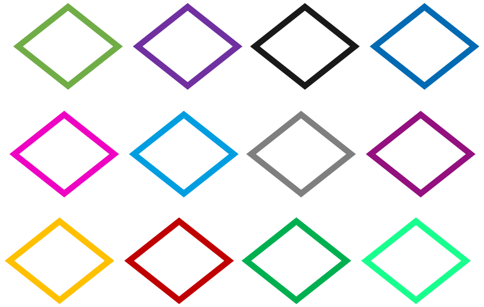
Prediction



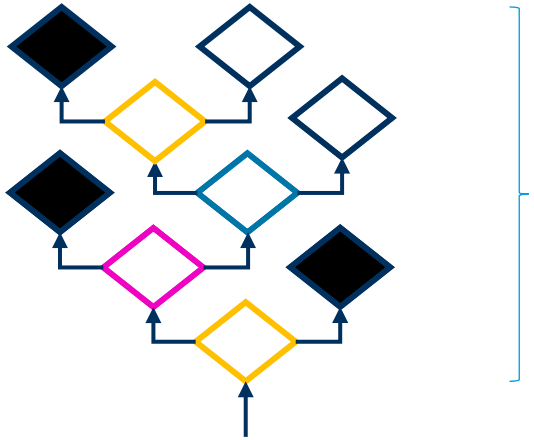
Random Forest Pixel Classifiers

Typical numbers for pixel classifiers in microscopy

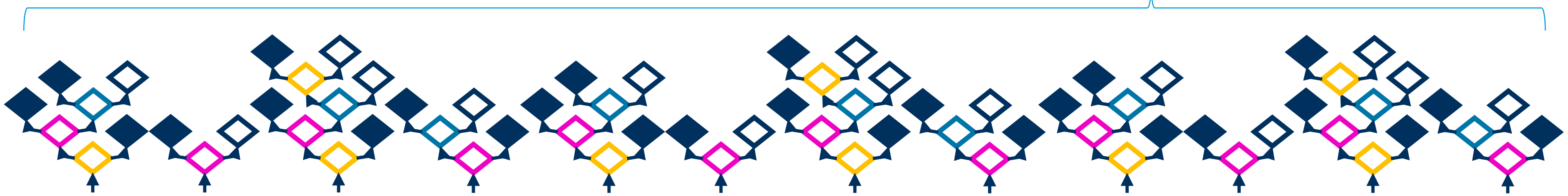
Available features:



- Gaussian blur image
- DoG image
- LoG image
- Hessian
- ....



Number of trees: > 100



Model validation

In order to assess model quality, we split the ground truth into two set

- Training set (50%-90% of the available data)
- Test set (10%-50% of the available data)

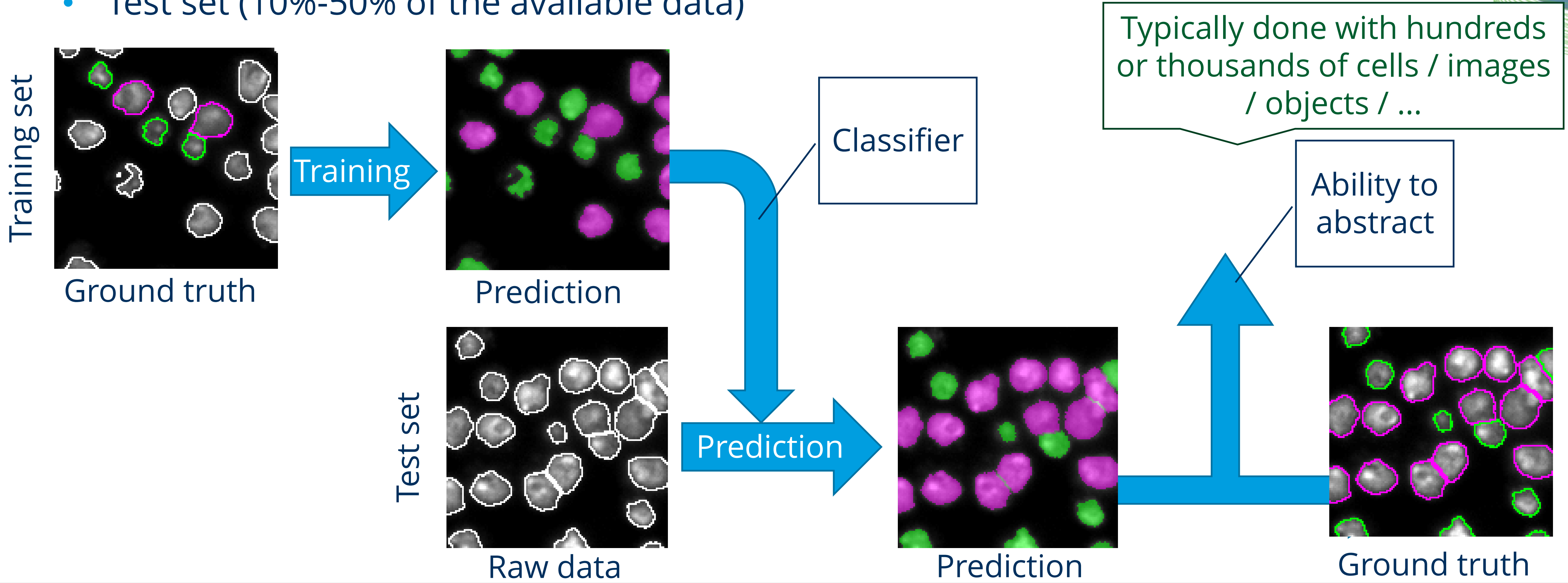
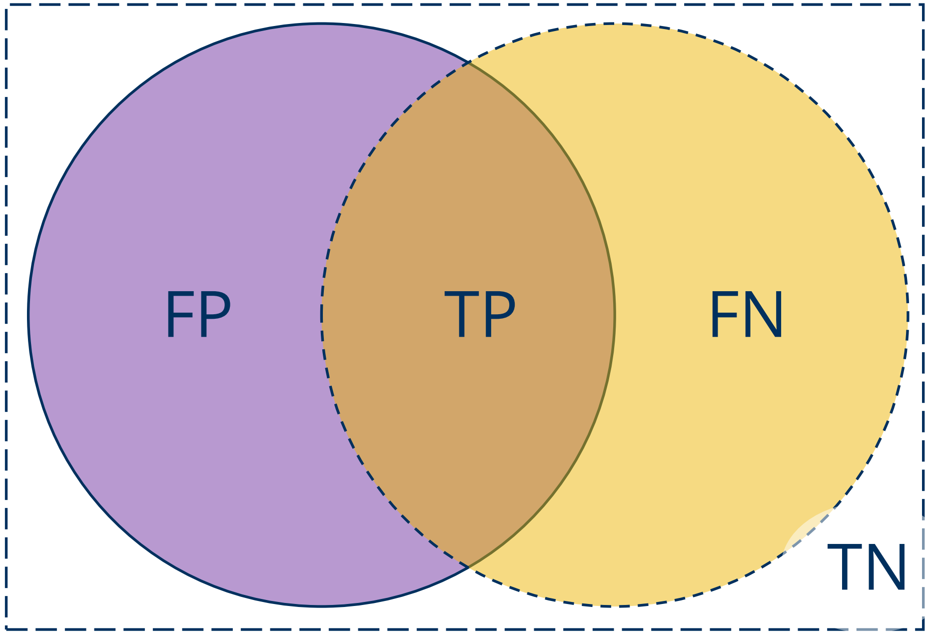


Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019)

Model validation

Based on the theory of sets



A — Prediction

B — Reference / ground truth

ROI — Region of interest

TP — True-positive

FN — False-negative

FP — False-positive

TN — True-negative

$$\text{Accuracy} = \frac{\text{correct classifications}}{\text{all classifications}}$$

This means:

$$= \frac{TP + TN}{FP + FN + TP + TN}$$

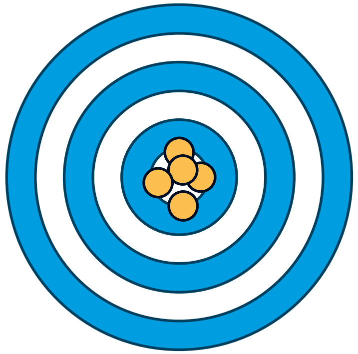
$$\text{Precision} = \frac{\text{Relevant retrieved instances}}{\text{All } \textbf{retrieved} \text{ instances}}$$

This *may* mean:

$$= \frac{TP}{FP + TP}$$

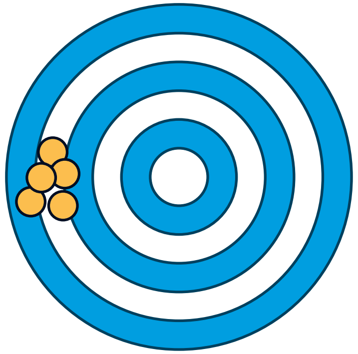
Read more:
https://en.wikipedia.org/wiki/Accuracy_and_precision
https://en.wikipedia.org/wiki/Precision_and_recall

Model validation: Accuracy versus precision

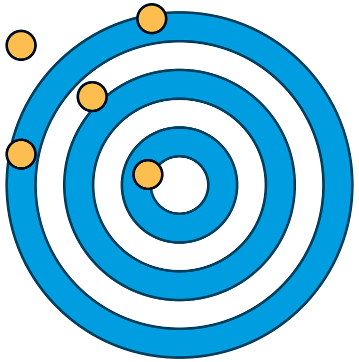


Accurate and precise

Accurate and but not precise



Not accurate and but precise

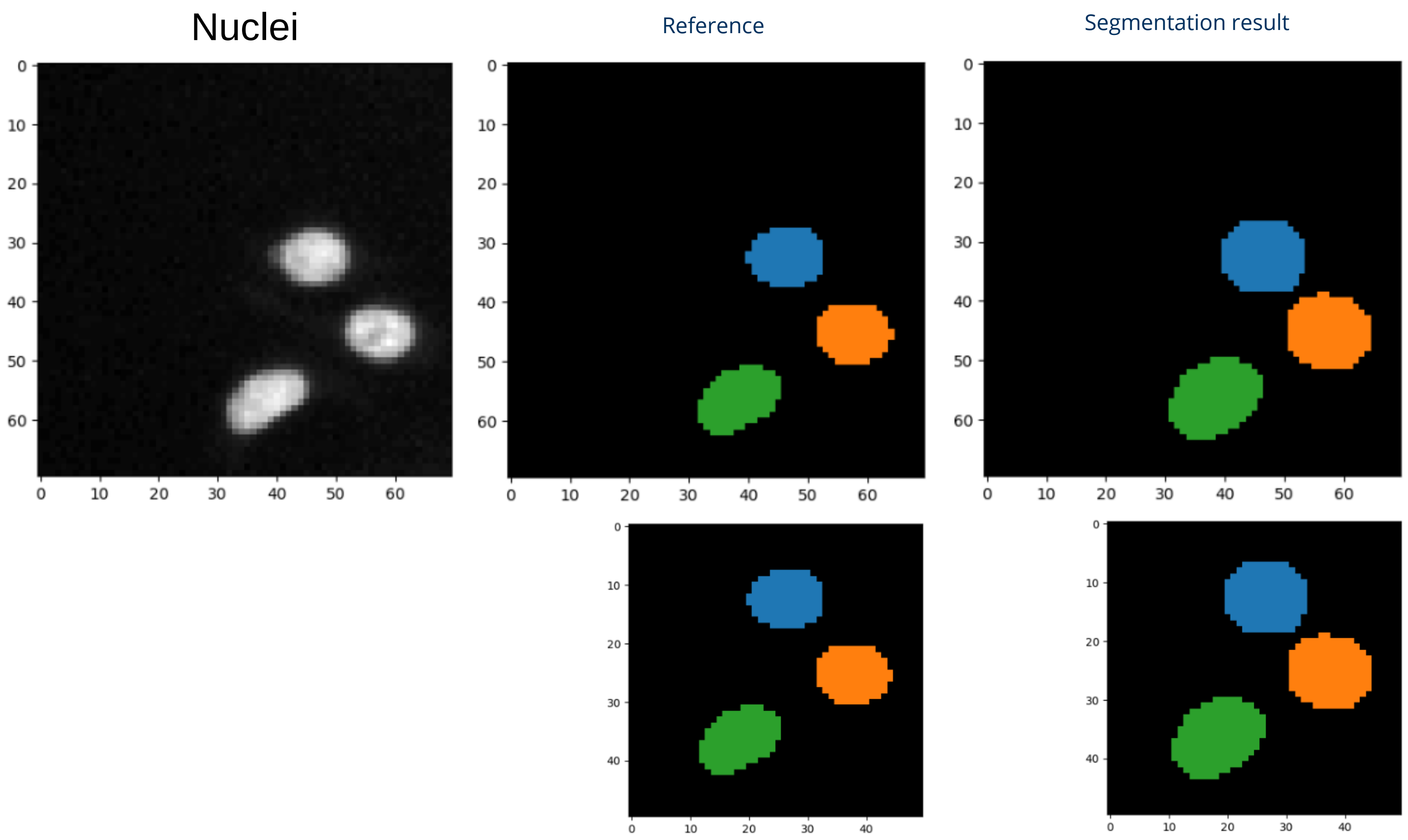


Neither accurate nor precise

Lesson learned:
A single quality metric cannot describe the whole situation

Model validation: Accuracy versus Jaccard Index

Side-effect of number of true negatives



$$A = \frac{TP + TN}{FN + FP + TP + TN}$$

$$J = \frac{TP}{FN + FP + TP}$$

Accuracy: 0.97
Jaccard Index: 0.73

Accuracy decreases because there are less correct black pixels (TN)

Accuracy: 0.95
Jaccard Index: 0.73

ScaDS.AI DRESDEN LEIPZIG

CENTER FOR SCALABLE DATA ANALYTICS AND ARTIFICIAL INTELLIGENCE

Deep Learning

Robert Haase

Funded by

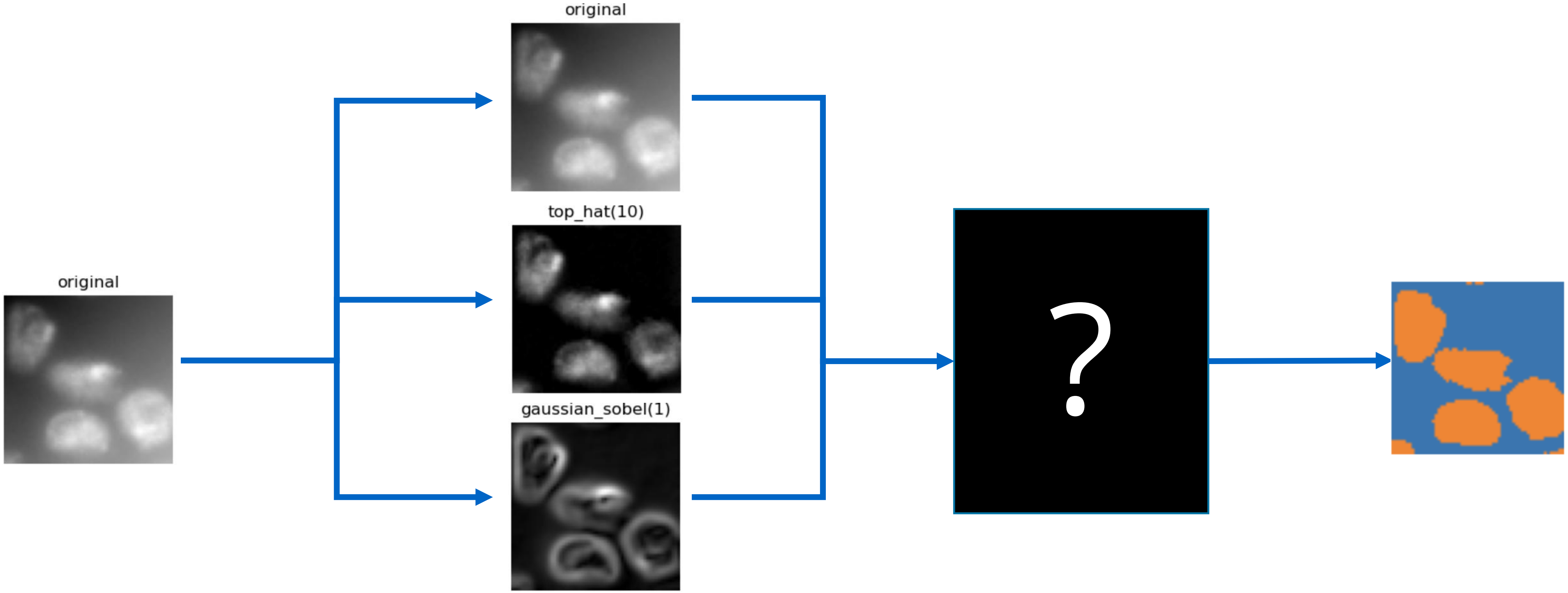
Bundesministerium für Bildung und Forschung

SACHSEN

Diese Maßnahme wird gefördert durch die Bundesregierung aufgrund eines Beschlusses des Deutschen Bundestages. Diese Maßnahme wird mitfinanziert durch Steuermittel auf der Grundlage des von den Abgeordneten des Sächsischen Landtags beschlossenen Haushaltes.

Machine learning for image analysis

In classical machine learning, we typically select features for training our classifier

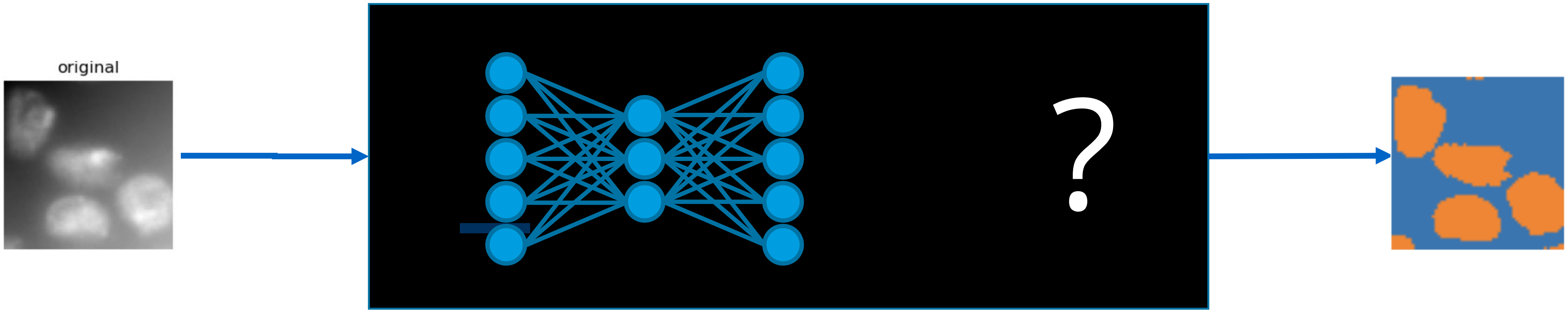


Features

Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019)

Deep learning for image analysis

In deep learning, this selection becomes part of the black box

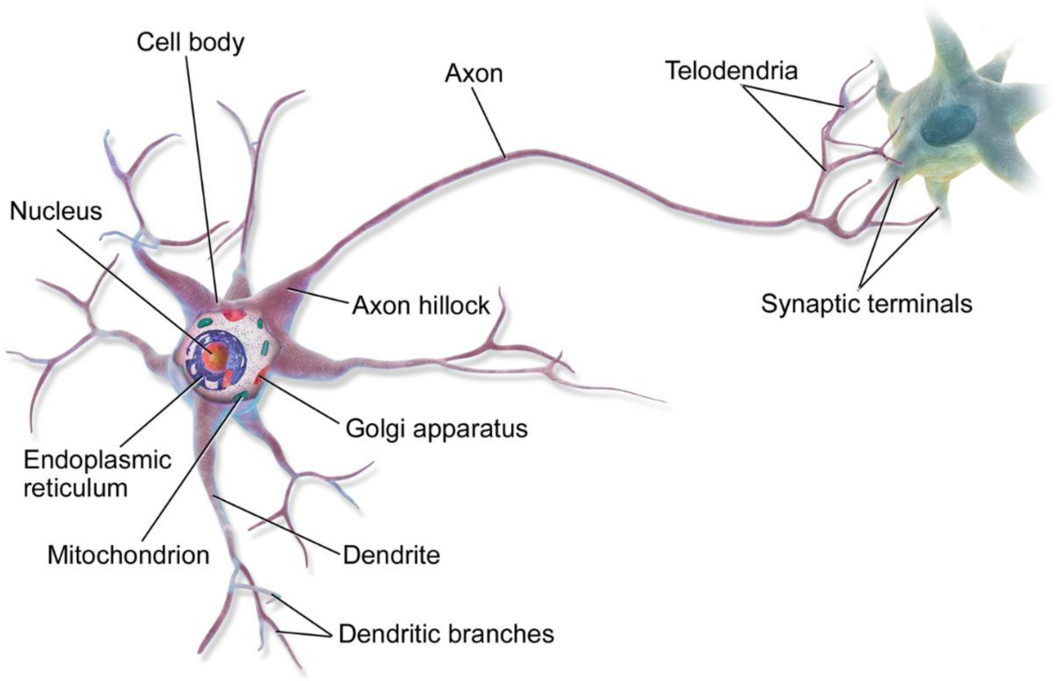


Neural networks

Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019)

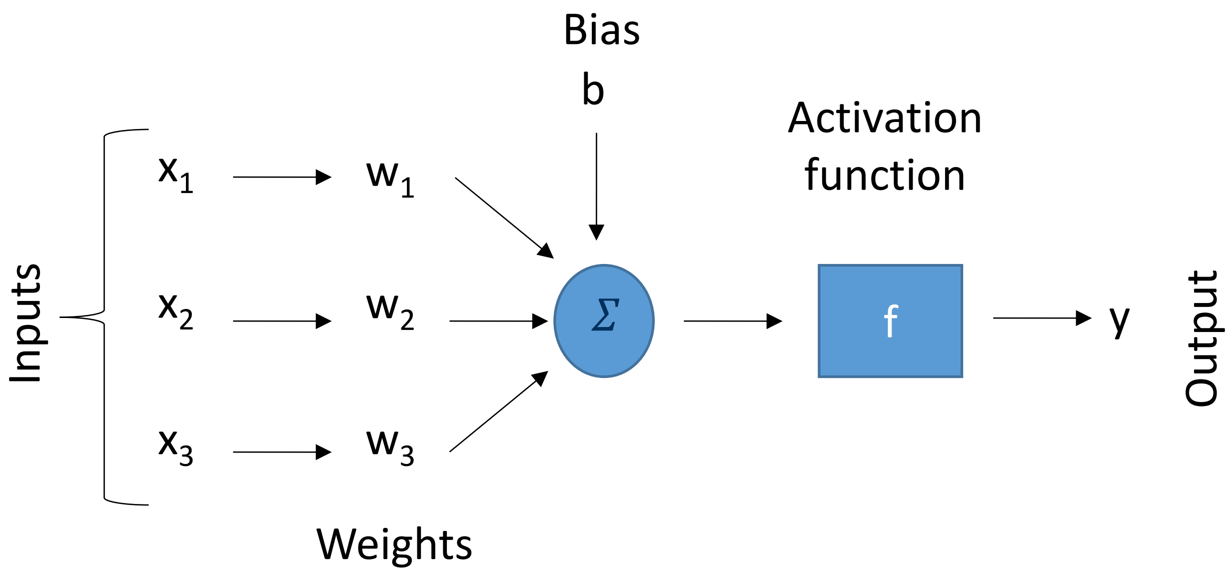
Neural networks

- How biologists see neurons



- How computer scientists see neurons

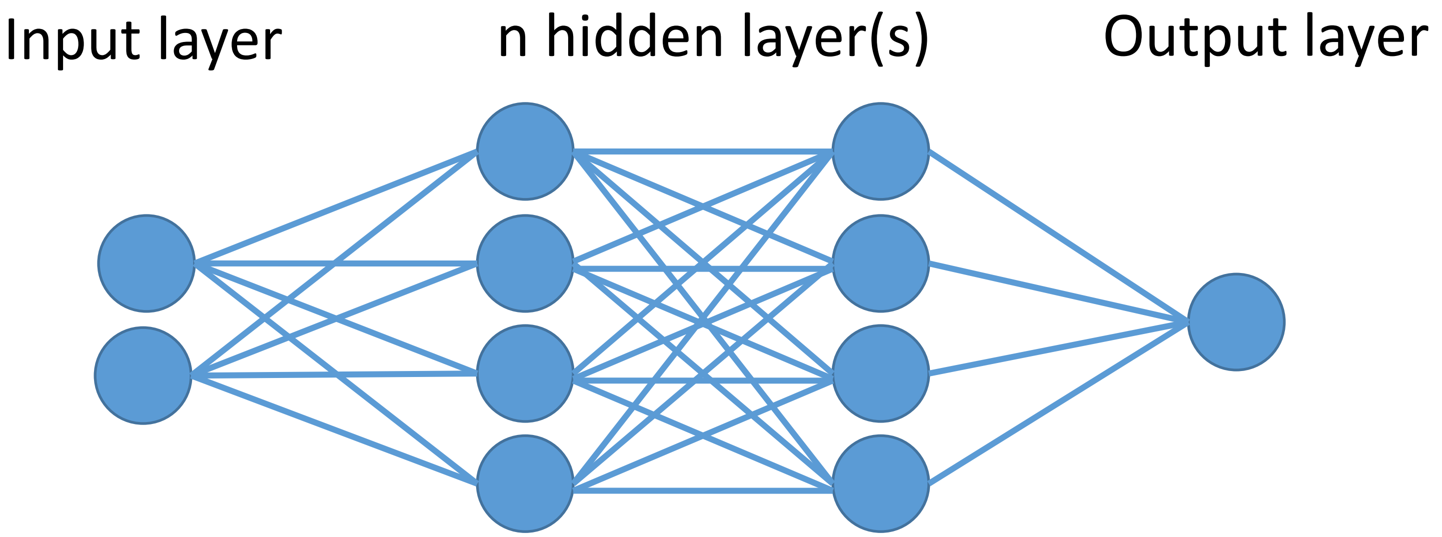
“perceptron”



Neural Networks

- Early form: “Multilayer Perceptron”
- fully connected class of feedforward artificial neural network

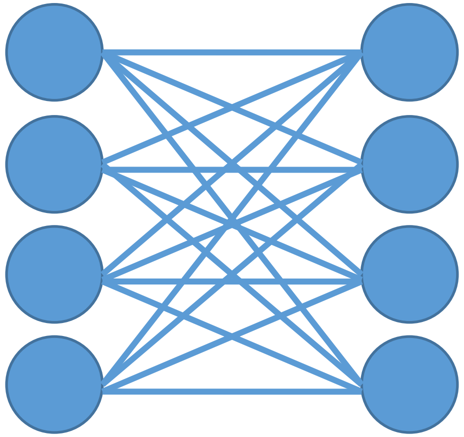
If there are *many* hidden layers, we speak of a *deep* neural network



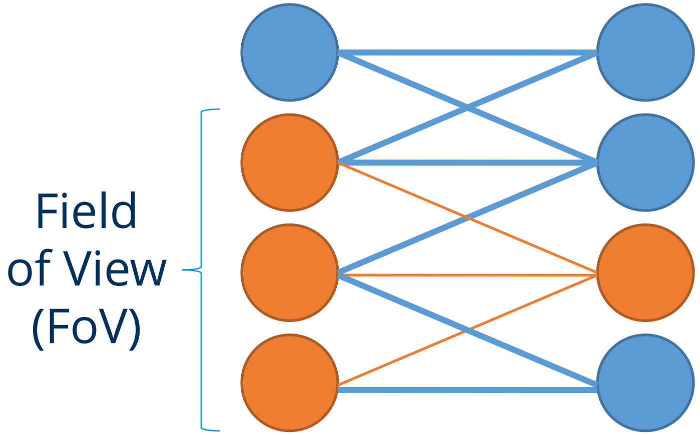
Convolutional neural networks

- Layer types

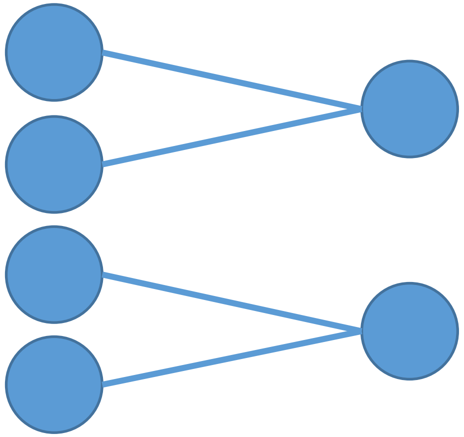
Fully connected layer



Convolutional layer



Pooling layer
("Max pool", "Average pool")



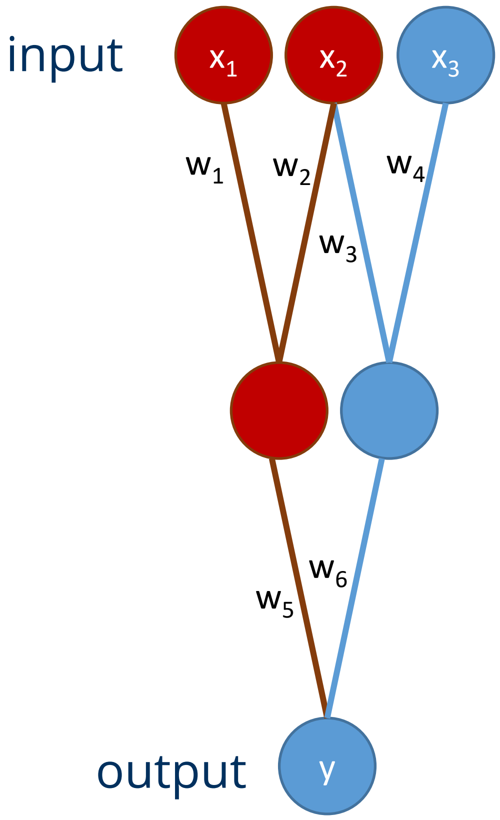
3	15	1	13
9	7	0	10
11	5	5	3
1	8	9	6

Max pooling

15	13
11	9

Convolutional neural networks

- Assuming we had no activation functions in the networklayers can be reduced by eliminating brackets!



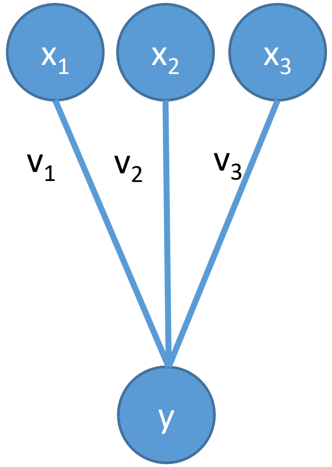
$$y = w_5(w_1x_1 + w_2x_2) + w_6(w_3x_2 + w_4x_3)$$

$$y = w_5w_1x_1 + w_5w_2x_2 + w_6w_3x_2 + w_6w_4x_3$$

$$y = w_5w_1x_1 +$$

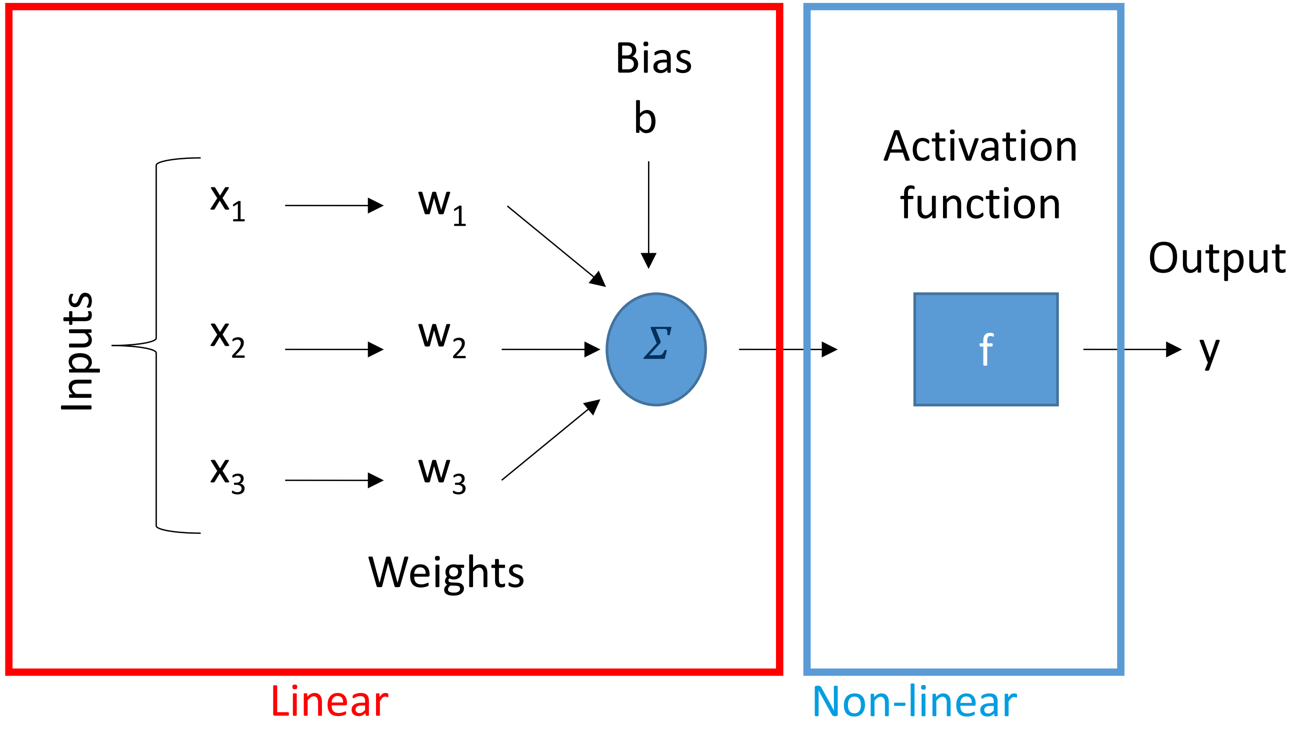
$$v_1 = w_5w_1$$
$$v_2 = w_5w_2 + w_6w_3$$
$$v_3 = w_6w_4$$

$$y = v_1x_1 + v_2x_2 + v_3x_3$$



Activation functions

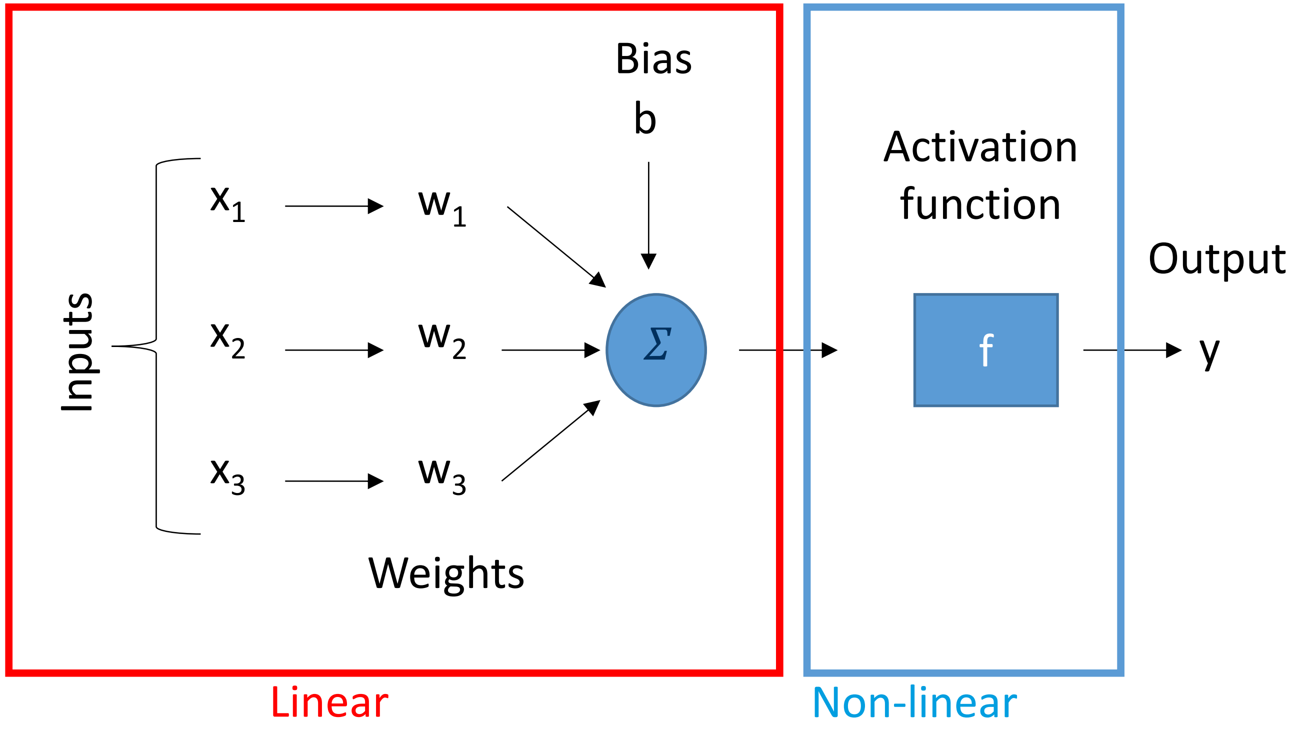
- Introduction of *non-linearity* and *activation functions* enabled what we call *deep-learning* today.

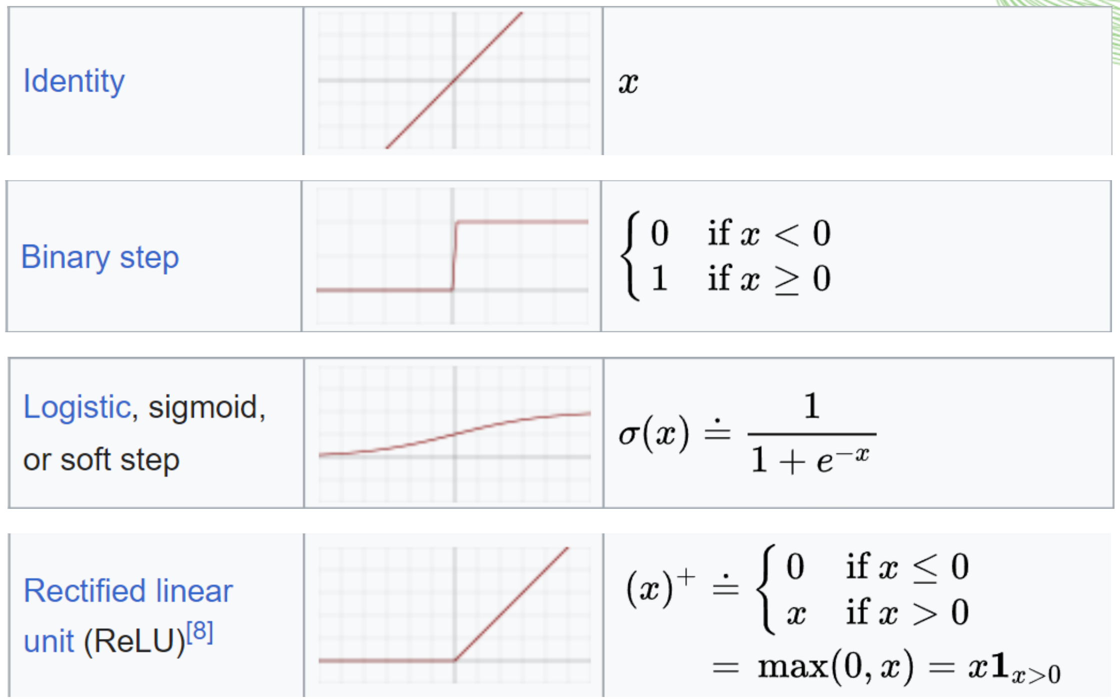


$$y = f(w_1x_1 + w_2x_2 + w_3x_3 + \mathrm{b})$$

Activation functions

- Introduction of *non-linearity* and *activation functions* enabled what we call *deep-learning* today.



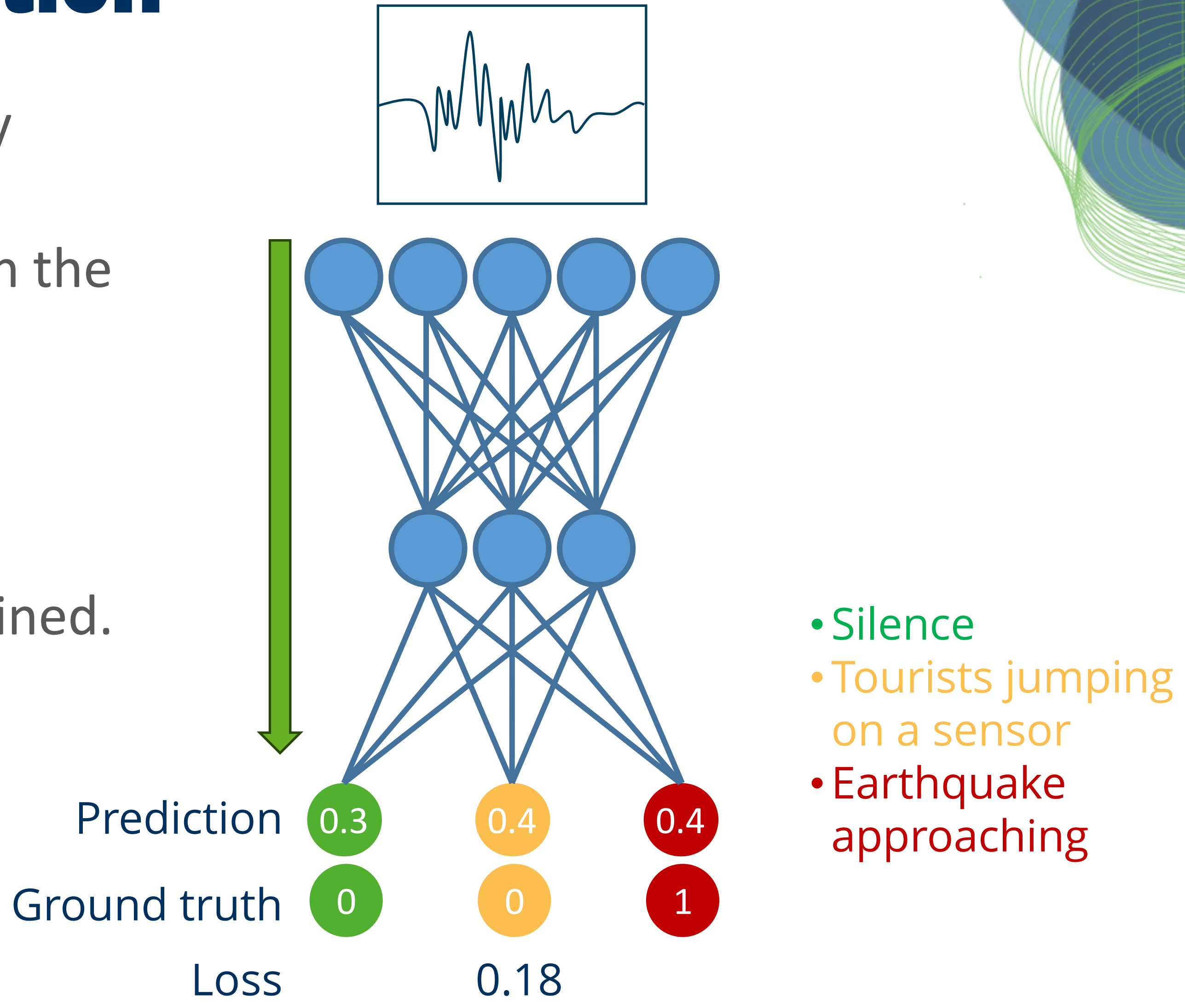


Identity		x
Binary step		$\begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$
Logistic, sigmoid, or soft step		$\sigma(x) \doteq \frac{1}{1+e^{-x}}$
Rectified linear unit (ReLU)[8]		$(x)^+ \doteq \begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases}$ $= \max(0, x) = x\mathbf{1}_{x>0}$

Source: https://en.wikipedia.org/wiki/Activation_function

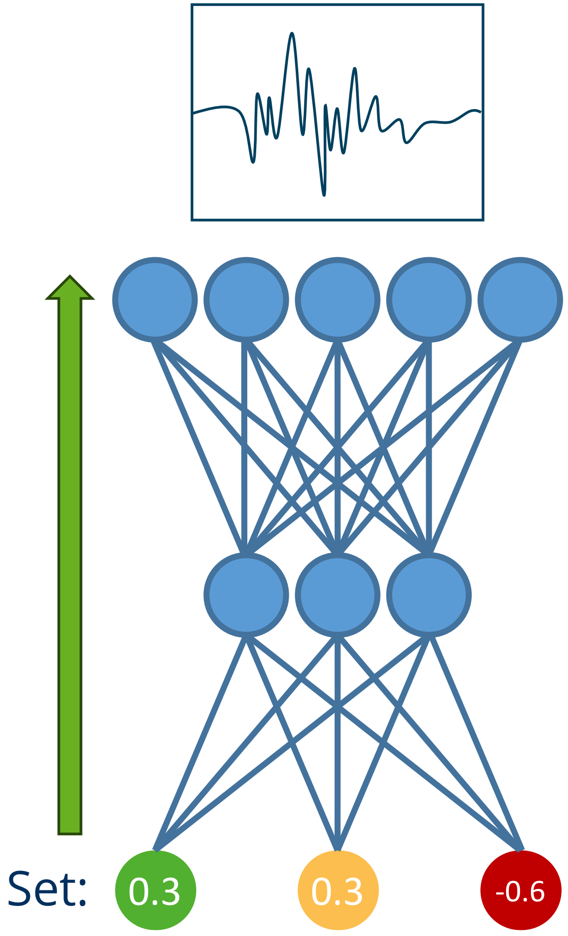
Learning: Back propagation

- Step 0: Initialize the network randomly (weights, bias)
- Step 1: Forward pass the input through the network, get an initial prediction
- Step 2: Compare the output with the ground truth, compute the error (loss function)
  - The loss function can be freely defined.
  - Example: mean squared error
- Step 3: Update weights



Learning: Back propagation

- Updating weights:
  - Set output to the error (per-parameter gradient)
  - Backward-pass: add/subtract gradients from weights, to push the network towards giving the right answer.
- Execute the same procedure for next sample
- Execute the same for multiple *epochs*



- Silence
- Tourists jumping on a sensor
- Earthquake approaching

Train- [validation]- Test-split

Training dataset (e.g. 80% of the data)

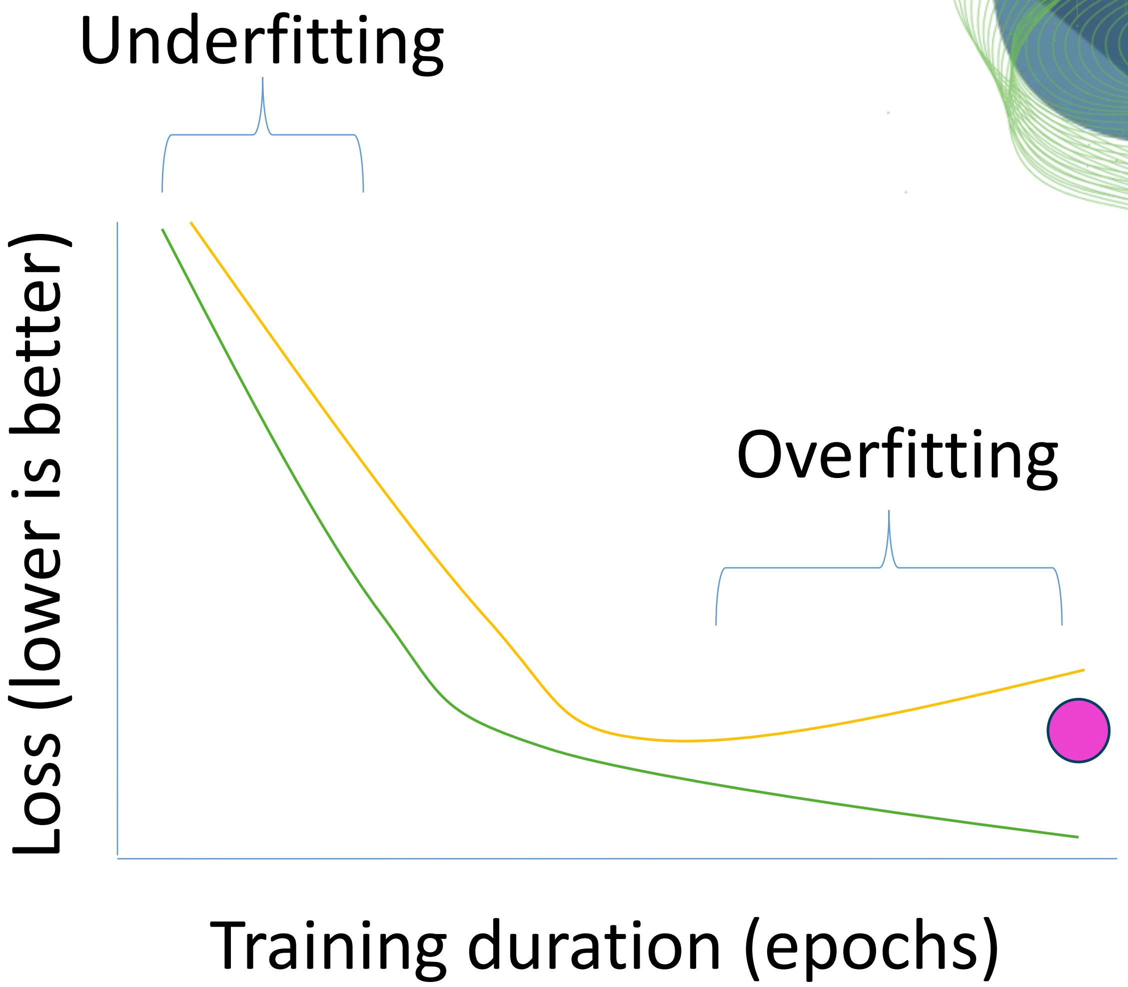
Used for training directly

Validation dataset (10% of the data)

After every iteration see if the model overfits

Test dataset (10% of the data)

Final evaluation after training is finished (once)

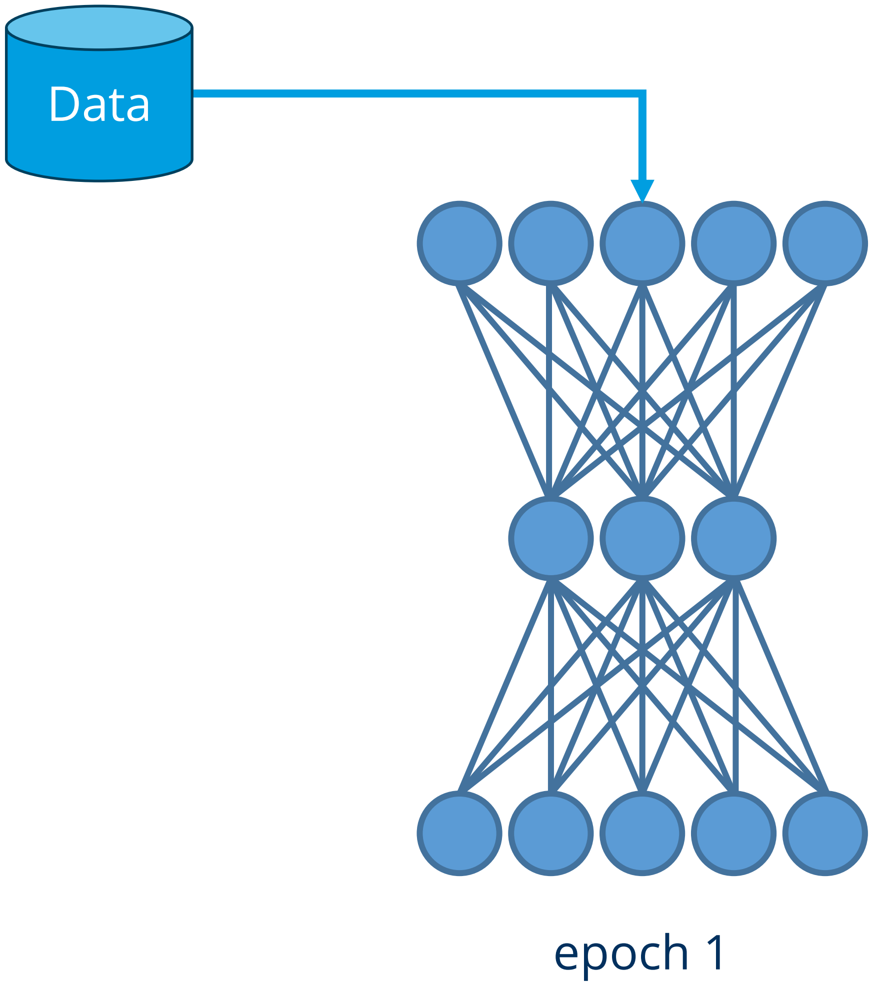


Training NNs: Batch size & epochs

Problem:

- Assume you have 10^{10} samples and attempt to train for 1000 epochs

-> 10^{13} backprop steps required.



Training NNs: Batch size & epochs

Problem:

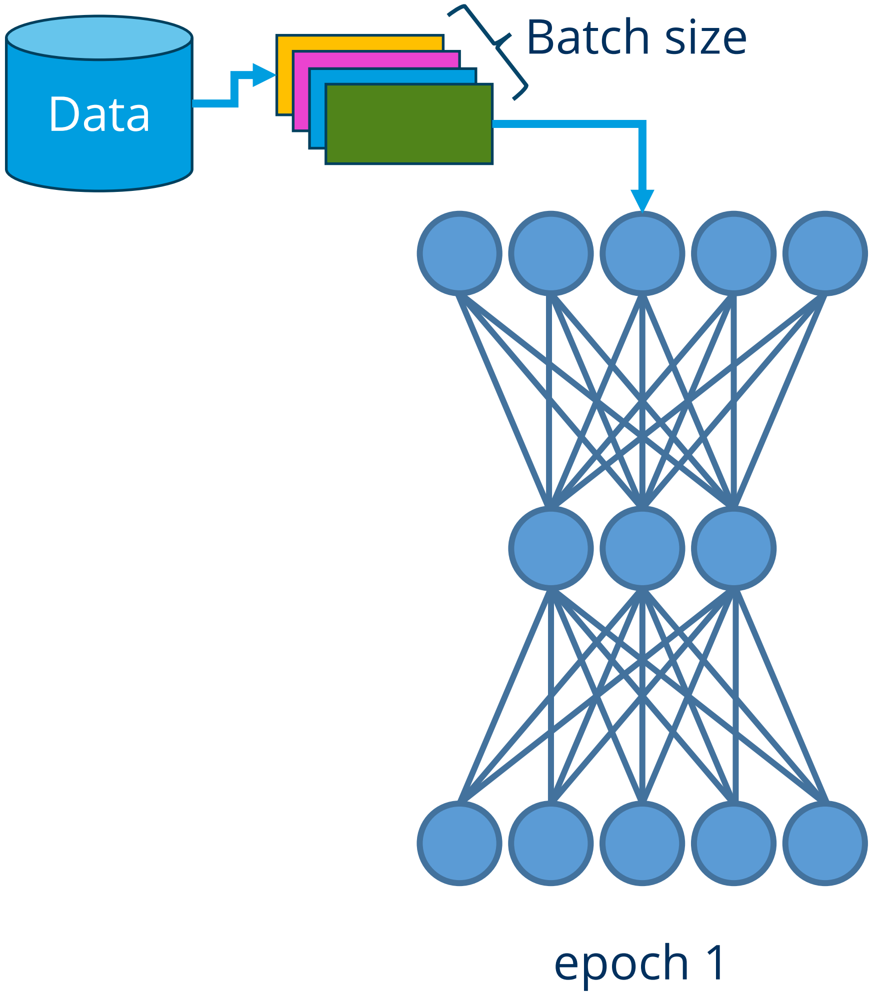
- Assume you have 10^{10} samples and attempt to train for 1000 epochs

-> 10^{13} backprop steps required.

Solution:

- Draw n=1000 random samples from the training data to train for one epoch.
- Next epoch: different n samples.

-> 10^{6} backprop steps required.



Training NNs: Batch size & epochs

Problem:

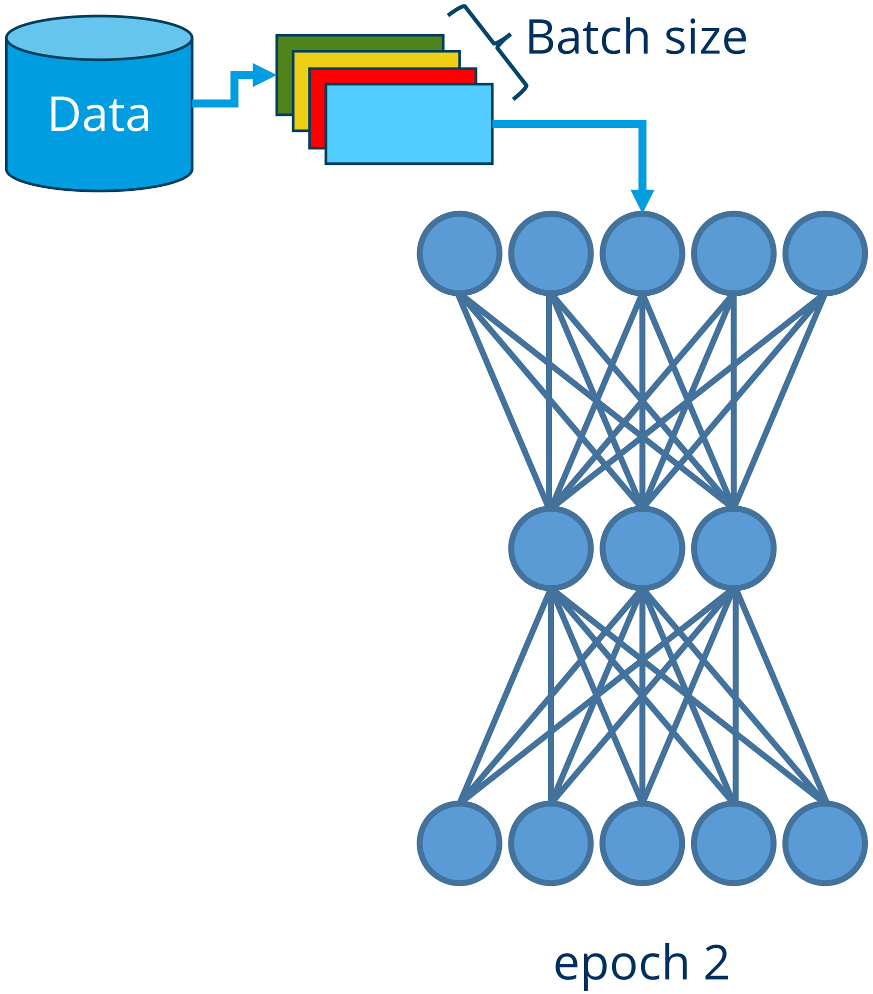
- Assume you have 10^{10} samples and attempt to train for 1000 epochs

-> 10^{13} backprop steps required.

Solution:

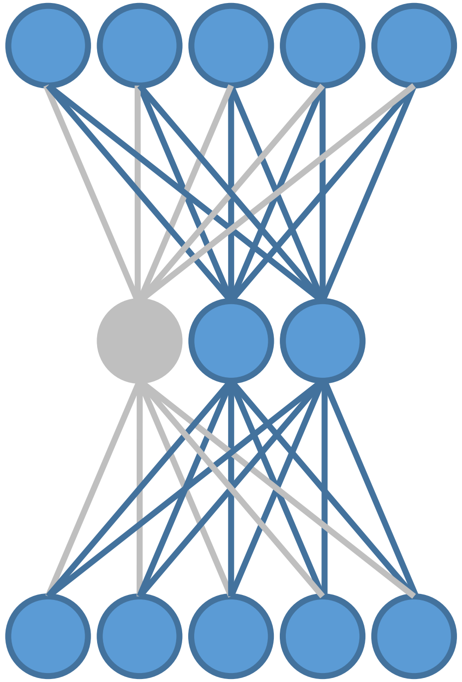
- Draw n=1000 random samples from the training data to train for one epoch.
- Next epoch: different n samples.

-> 10^{6} backprop steps required.



Training NNs: Drop-out

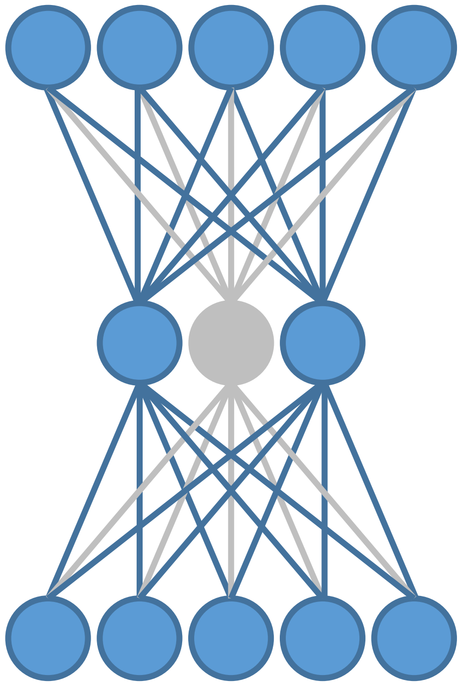
- Drop-out: deactivating individual neurons during training
- Helps with over-fitting, because the network cannot rely on individual neurons by chance being well trained, while others remain randomly initialized
- Example: drop-out-rate: 30%



epoch 1

Training NNs: Drop-out

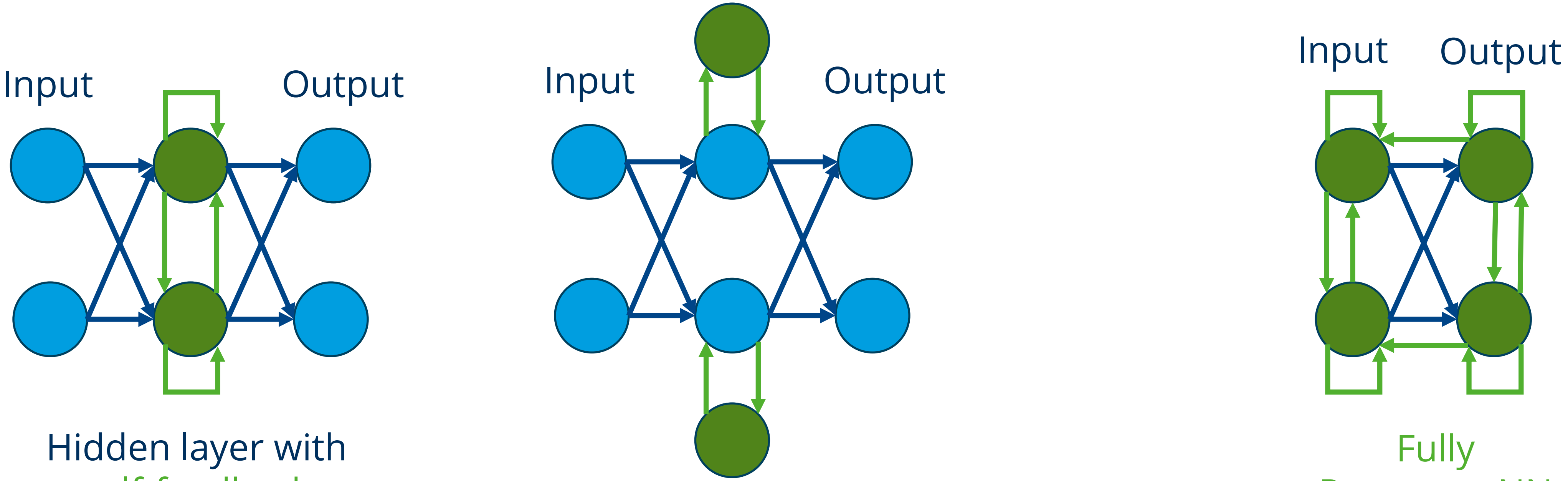
- Drop-out: deactivating individual neurons during training
- Helps with over-fitting, because the network cannot rely on individual neurons by chance being well trained, while others remain randomly initialized
- Example: drop-out-rate: 30%



epoch 2

NN Architectures: Recurrent Neural Networks

Introducing some form of memory through additional connections and nodes.



Hidden layer with self-feedback

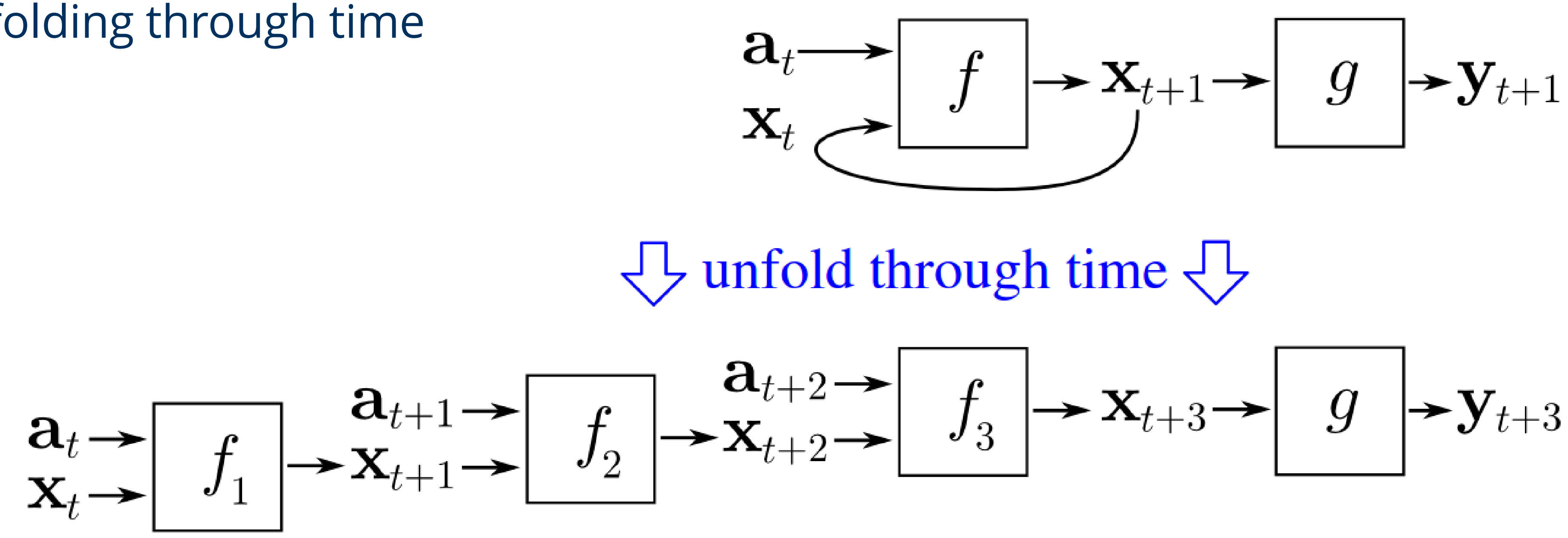
Hidden layer with context nodes (Elman network, 1990)

Fully Recurrent NN

Figure sources: Staudemeyer & Rothstein Morris 2019
https://arxiv.org/pdf/1909.09586

Training Recurrent Neural Networks

- Backpropagation through time
- Computationally expensive
- Unfolding through time



https://en.wikipedia.org/wiki/Backpropagation_through_time

NN Architectures: Long Short-Term Memory (LSTM)

Differentiation between updating short-term memory (all the time) and updating long-term memory ([not] forgetting) thanks to separate input- and forget-gates.

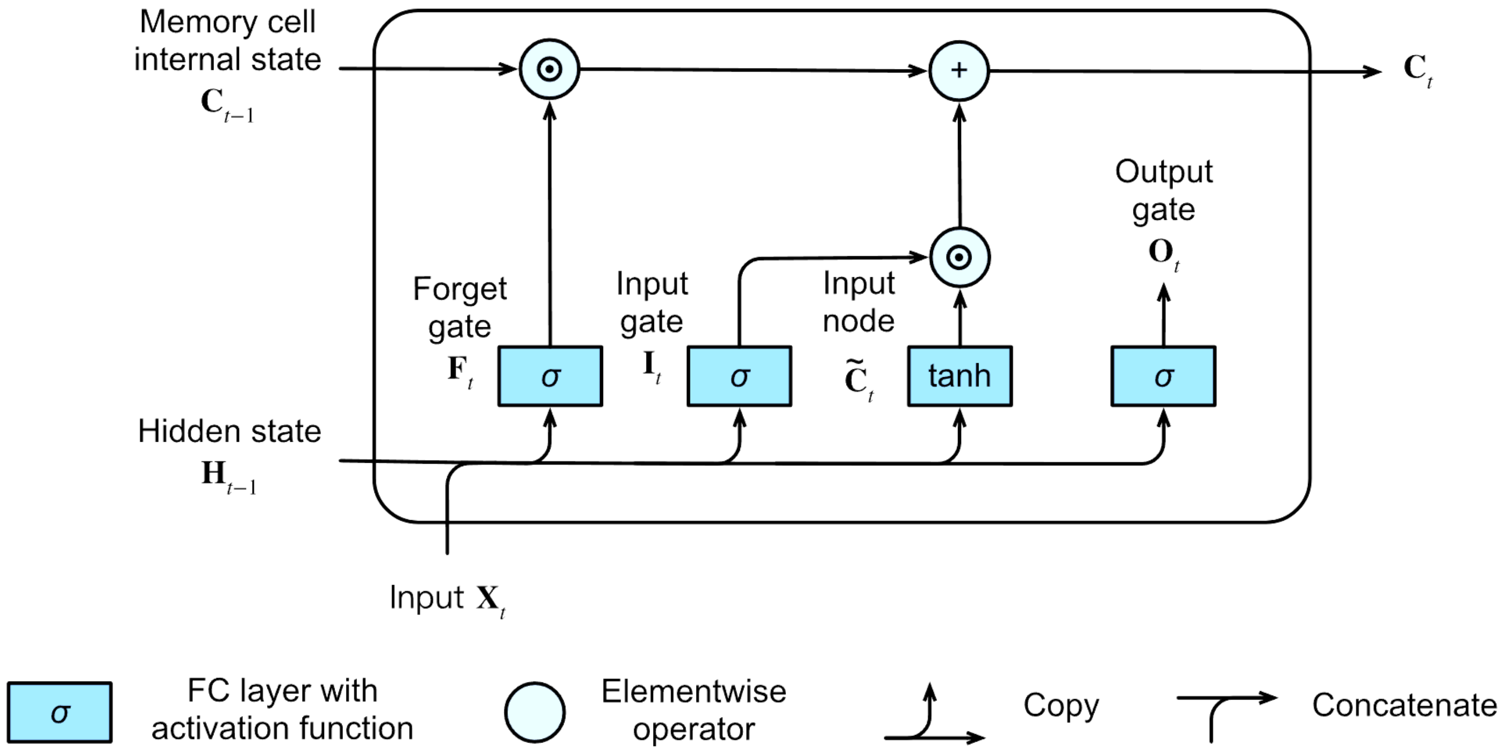
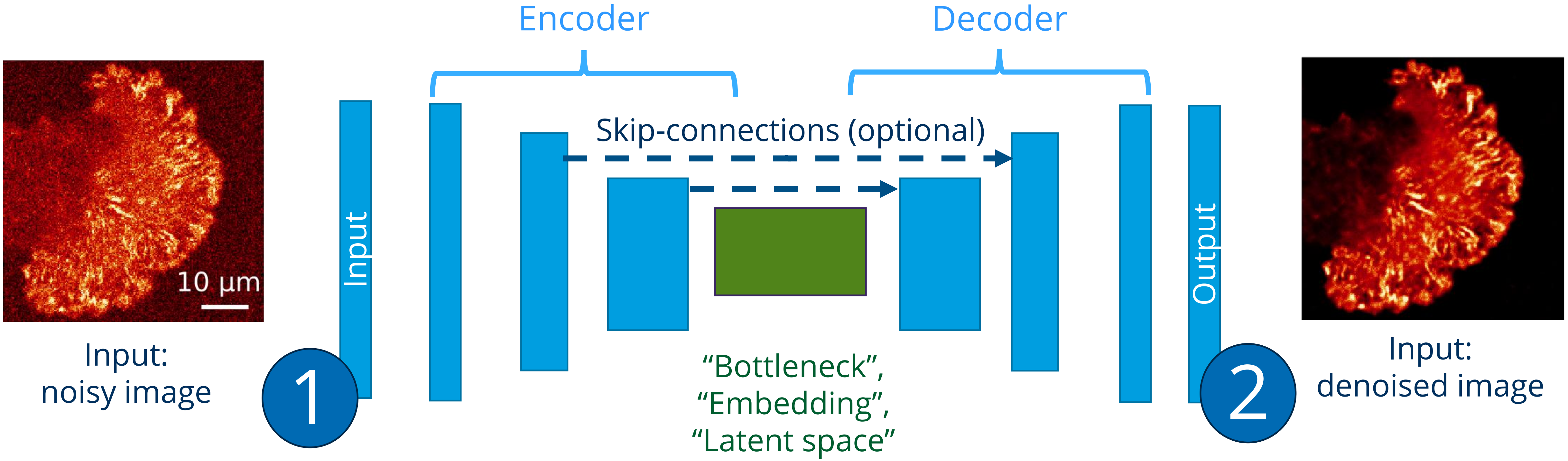


Figure adapted from: D2AI, licensed CC-BY-SA 4.0 https://d2l.ai/chapter_recurrent-modern/lstm.html
Read more: Hochreiter and Schmidhuber (1997). https://direct.mit.edu/neco/article-abstract/9/8/1735/6109/Long-Short-Term-Memory

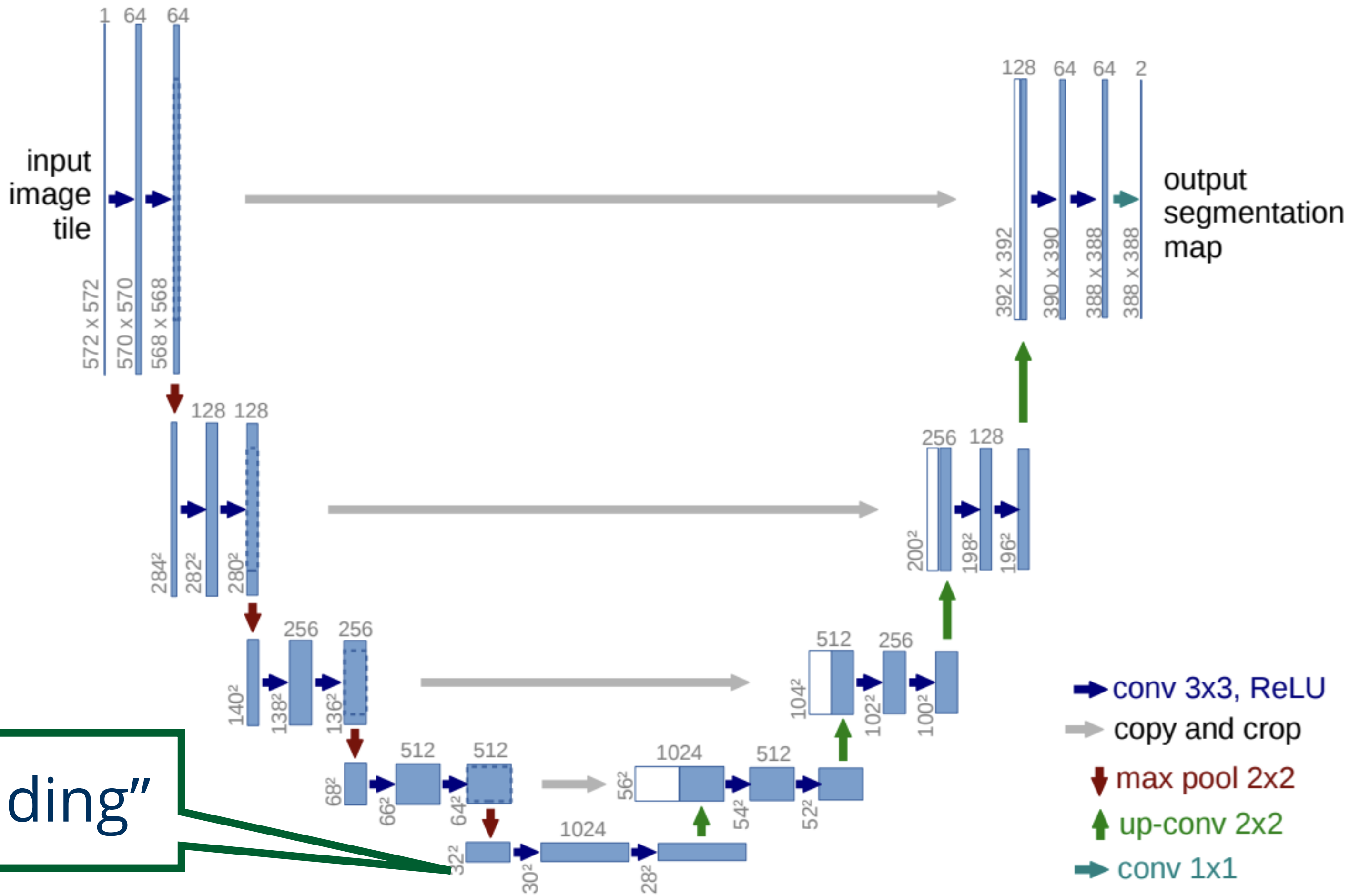
Traditional architecture: Encoder-Decoder Networks

Related: „Auto-encoder“, „Variational Auto-Encoder“, „U-Net“



Images demonstrating noise2void, cropped from https://github.com/HenriquesLab/ZeroCostDL4Mic/blob/master/Wiki_files/FigureS2.png license: MIT

Traditional architecture: Encoder-Decoder Networks



"Embedding"

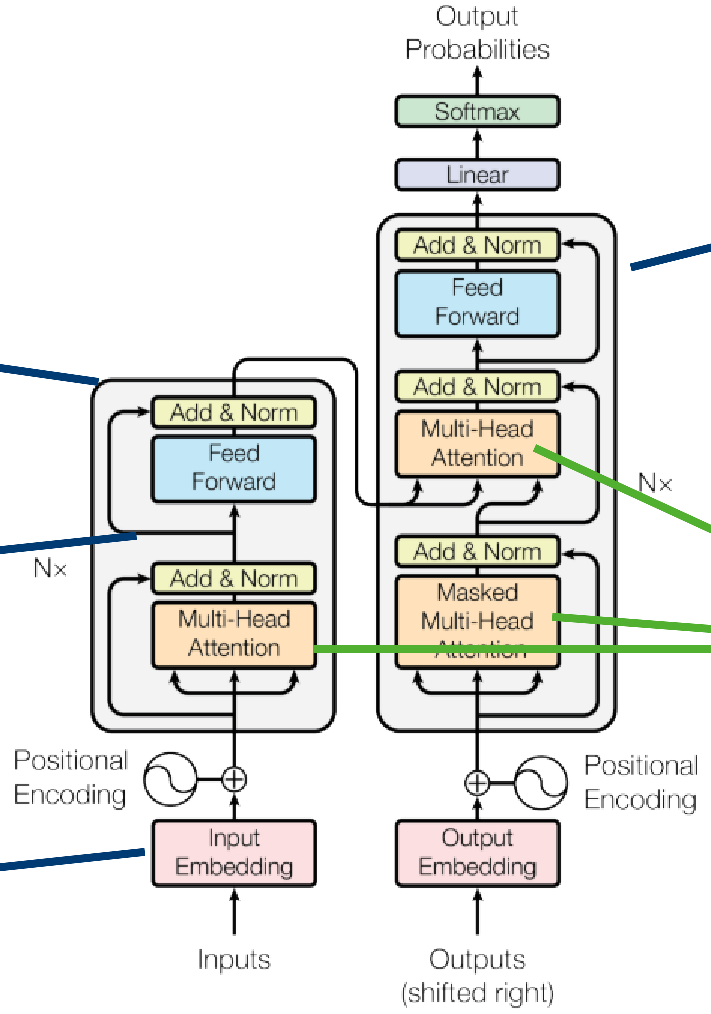
Figure source: Ronneberger et al 2015
https://arxiv.org/pdf/1505.04597

NN Architectures: Transformers

Encoder

Vectors and matrices

Embedding



Decoder

Attention

Source: Vaswani et al (2017)
https://arxiv.org/abs/1706.03762

Scaled dot-product attention

Attention score: How much related are two words?

Query: For which word are we calculating attention?

Key: To which word are we calculating attention

Value: Relevance of the query-key relationship

The cat is black and white.

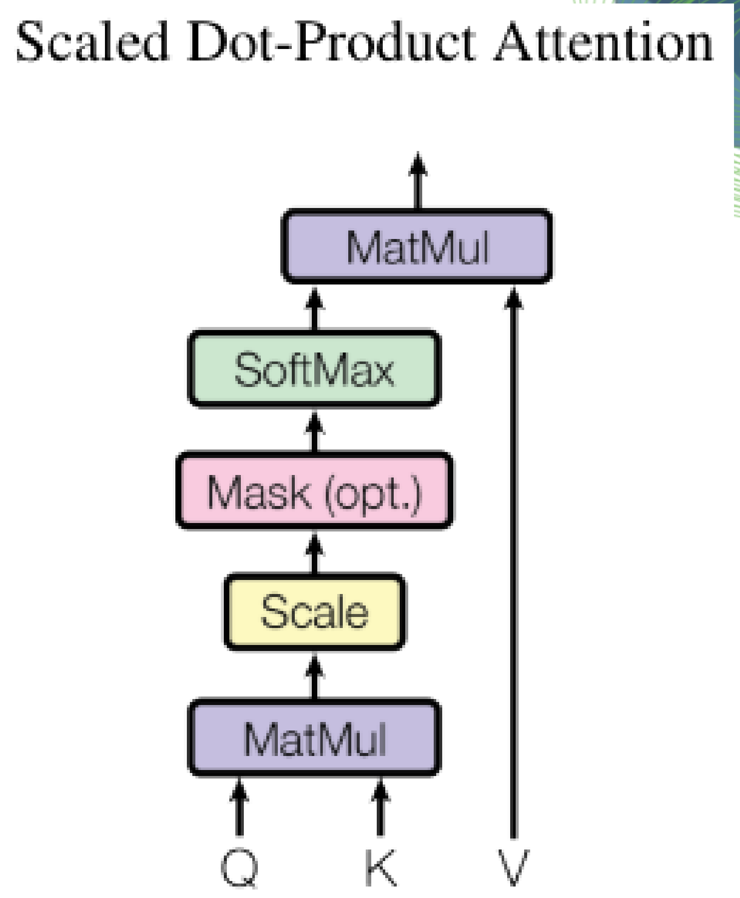
Relevance value: 0.1

attention score

The cat is meowing.

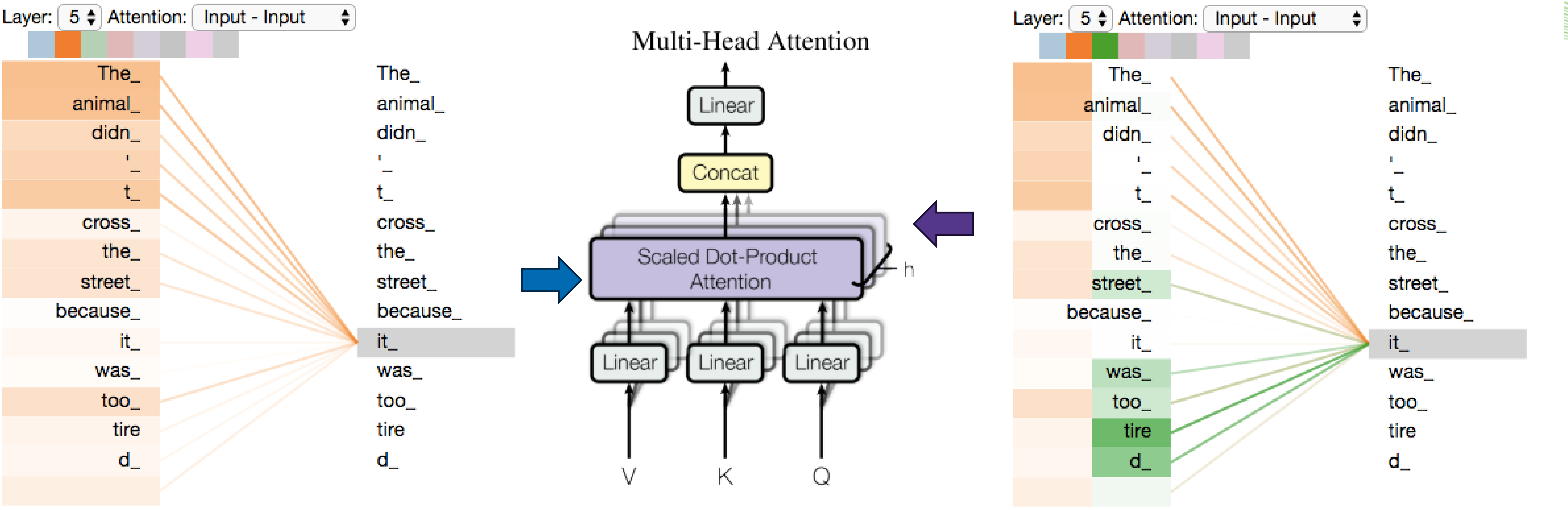
Relevance value: 0.9

attention score

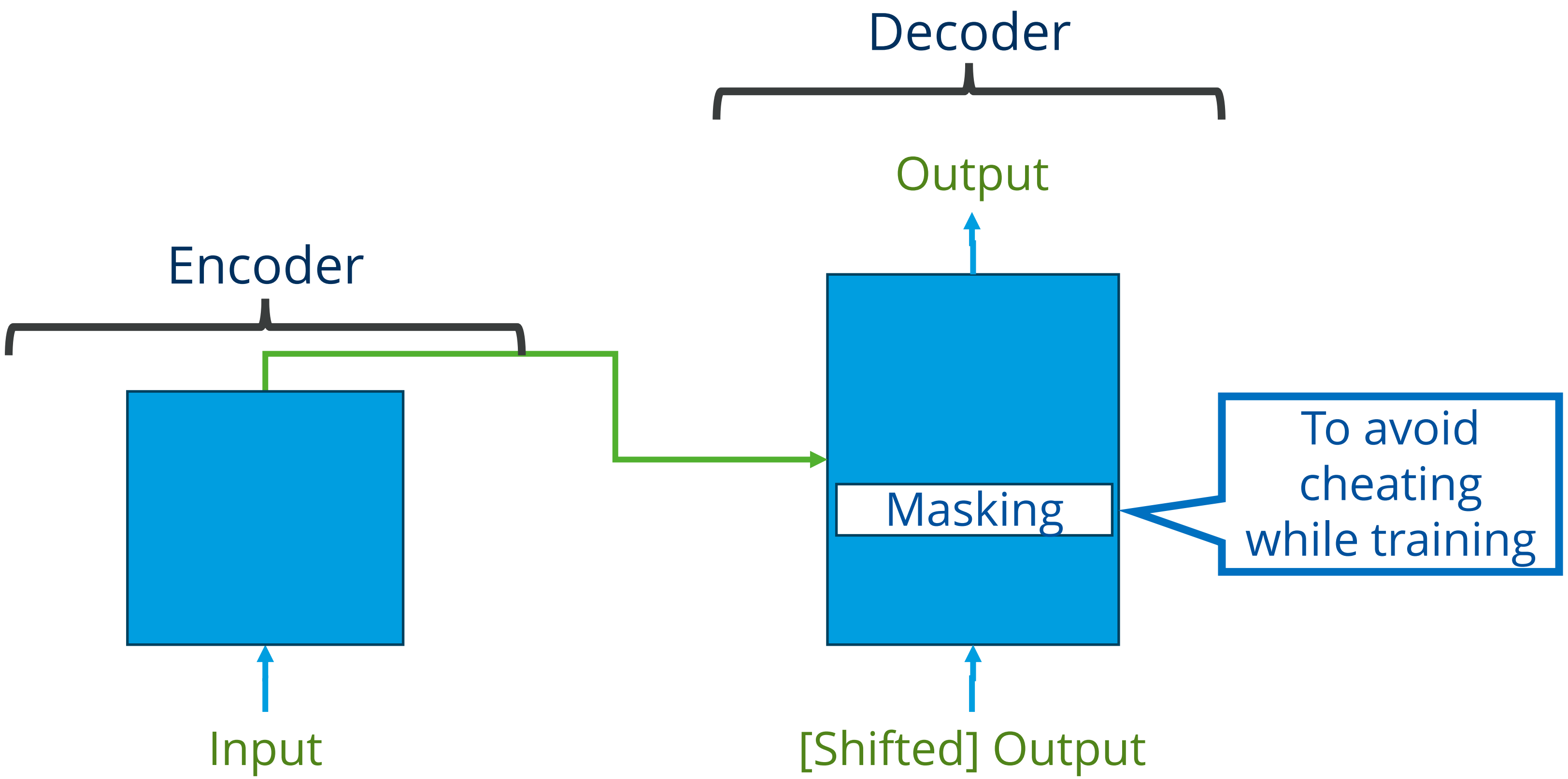


Multi-head attention

Multiple aspects represented by multiple attention heads



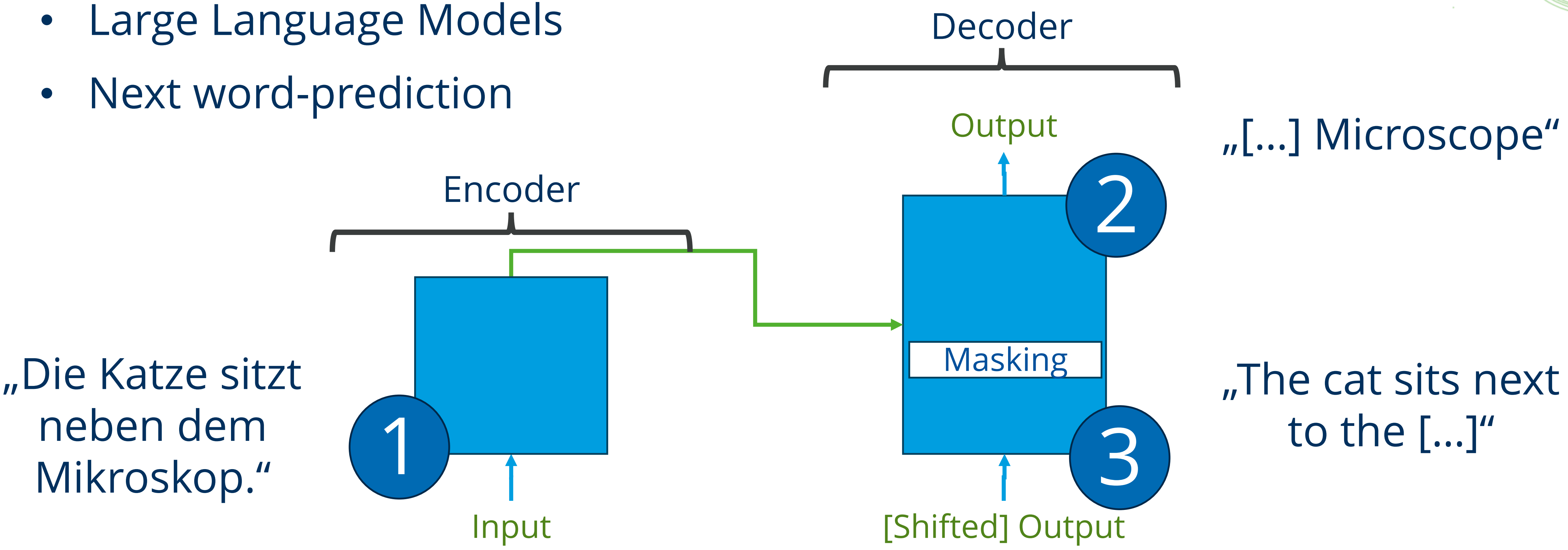
NN Architectures: Transformers



NN Architectures: Transformers

Related terms:

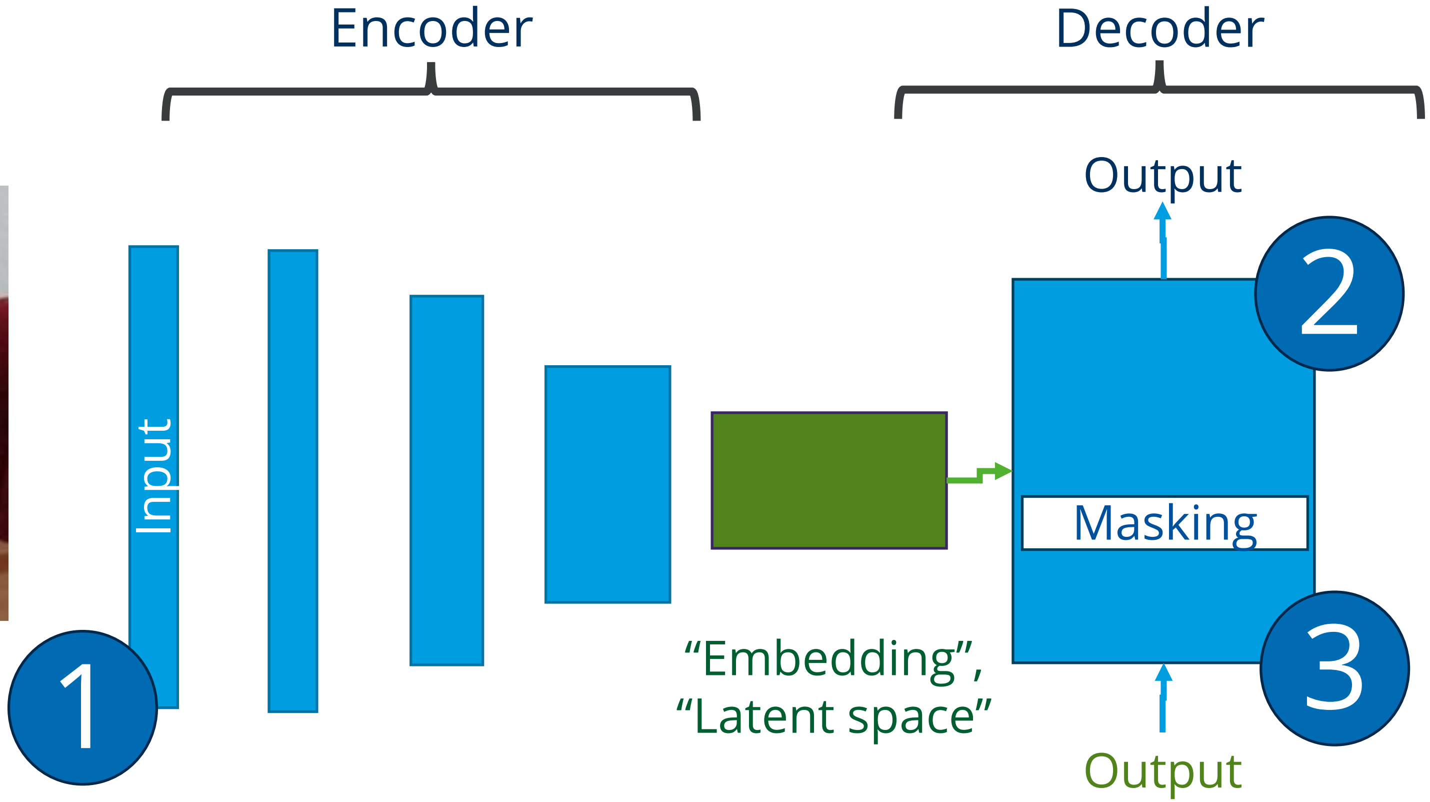
- Generative Pretrained Transformer (GPT)
- Large Language Models
- Next word-prediction



NN Architectures: Vision Language Models

VLMs use combinations of traditional neural network architectures and transformers.

Input image

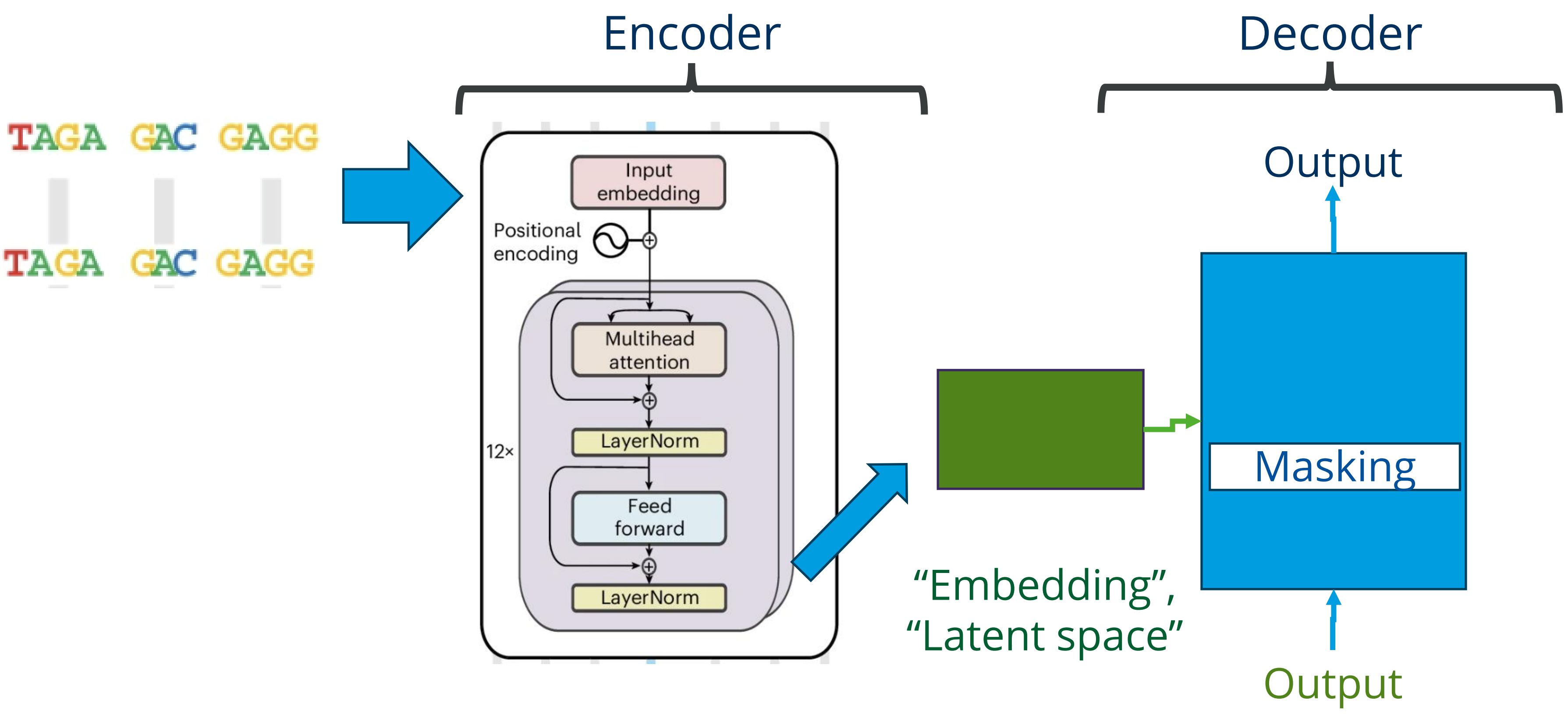


„[…] Microscope"

„The cat sits next to the […]"

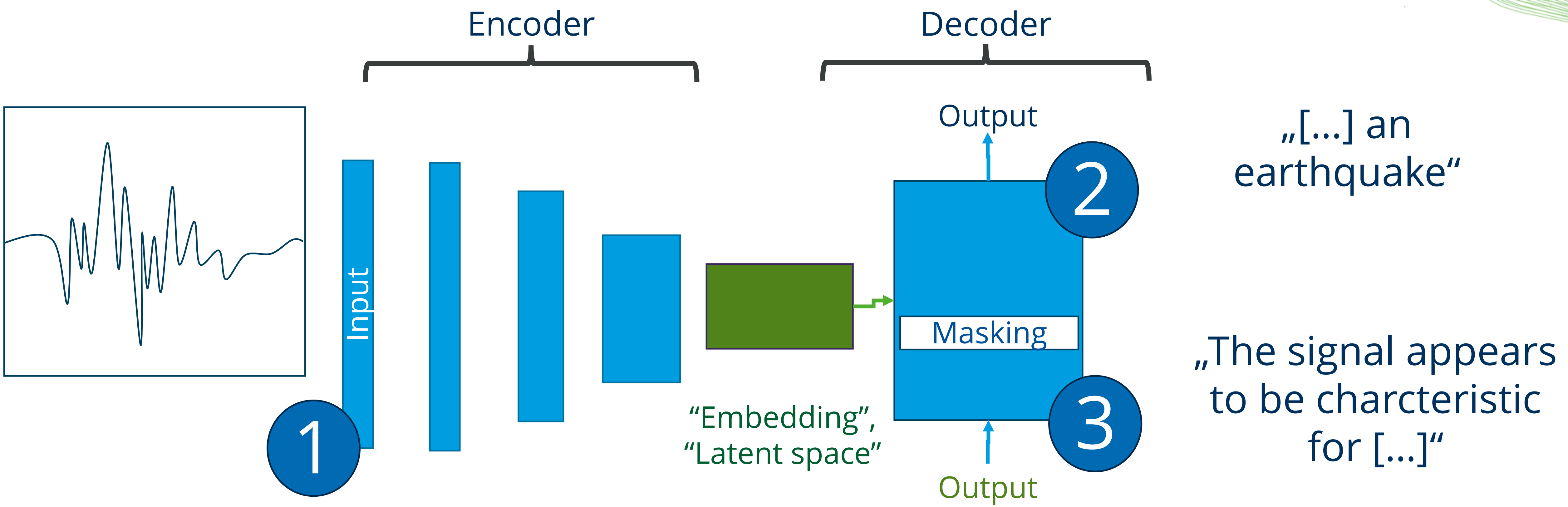
NN Architectures: DNA Language Models

DNA-LMs use a variation of the transformer architecture.



Multi-modal Language Models

MMLMs use combinations and/or variations of traditional neural network architectures and transformers.

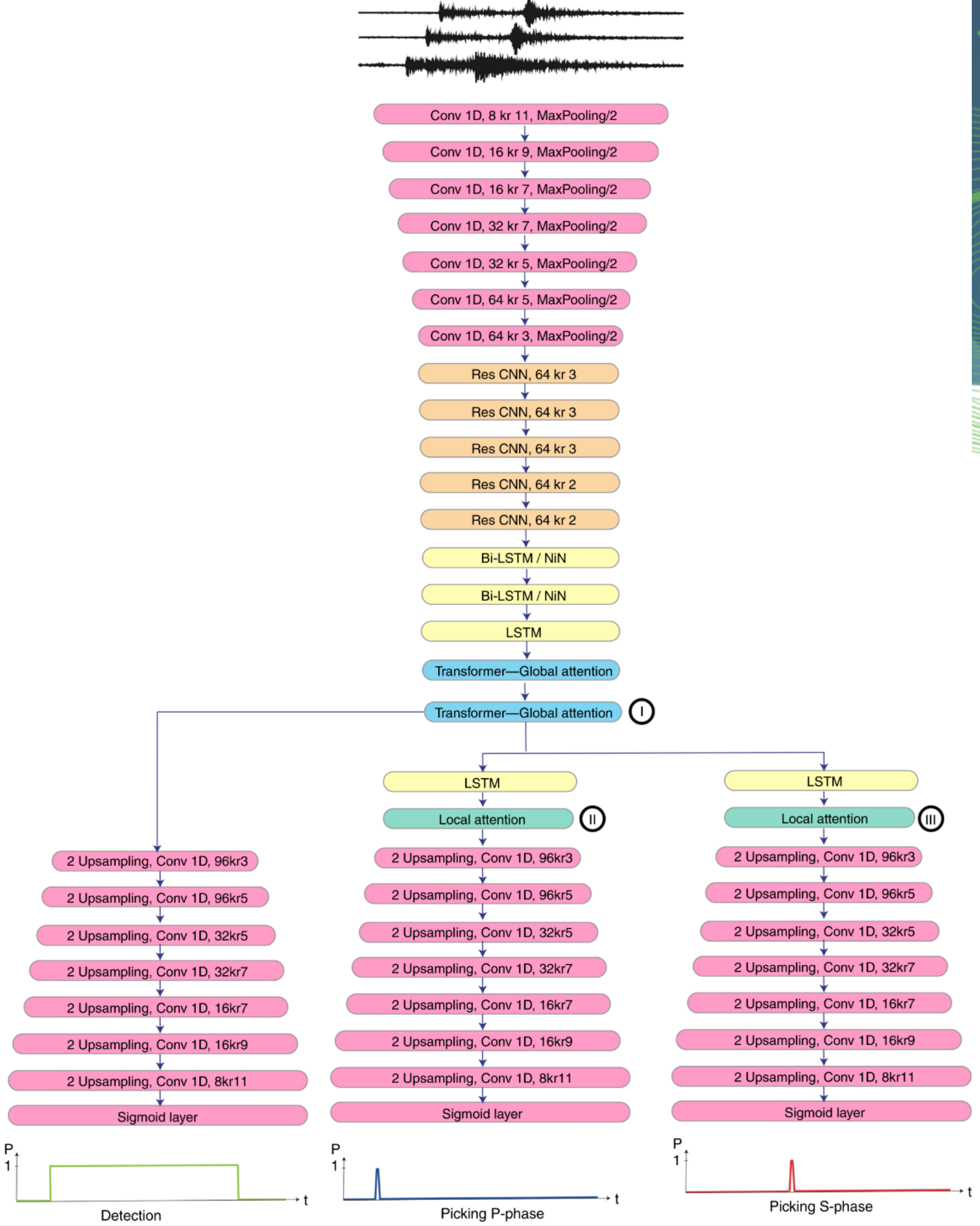


NN Architectures

Modern NN architectures combine techniques quite freely. Example, for large earthquake detection:

- LSTMs
- Transformers
- Convolutional
- Attention

Combining architectures sometimes appears *more art than science*. Computer scientists world-wide struggle comparing different architectures.



Summary

Unsupervised ML: Explorative data science, Embeddings

Supervised ML / DL: Preduction: classification / regression, Embeddings

Explainability: SHapleys Additive exPlanations (SHAP-Analysis)

Neural networks

- Many hidden layers -> *deep* learning, Embeddings
- Training: Drop-out, batch-size, epochs
- RNNs / LSTMs -> Memory
- Transformers -> Attention, Embeddings

Good scientific practice

- Train-test-split
- Overfitting / underfitting