Crash Course on Machine Learning for Chemists

Bernd Ensing,
Robert Pollice

Part I
An Introduction to AI
15:00 – 15.40

Part II
Applied AI in Chemistry
15:40 – 16.20

Part III
Exam / Quiz
16:20 – 17.00

Part I: An Introduction to AI

Learning Objectives

1. Understand the overall workflow of machine learning
   - dealing with data
   - choosing a model
   - training and validation
   - inference; applying machine learning models

2. Acquaint yourself with the fancy jargon
   (to bluf your way into the machine learning collective)

3. Prepare you to work with colleagues / students on machine learning

4. Become a **<u>certified</u>** Crash Course master ← if you manage to pass the crash course exam-quiz

5. *Learning objective for BE & RP: how useful is a crash course?*
   Follow-up? Feedback needed!

What is Machine Learning and AI?

a super brief history in 15 milestones

2020

2000

1980

1960

1940

Present - Generative AI (e.g., ChatGPT, DALL·E) and multimodal models redefine AI's applications across industries and research fields.

2021 - **AlphaFold** by DeepMind solves the decades-old protein folding problem, transforming structural biology.

2020 - **GPT-3**, a large language model, showcases state-of-the-art natural language generation capabilities.

2017 - Introduction of **transformer models** (e.g., Attention Is All You Need) revolutionizes natural language processing.

2016 - Google DeepMind's **AlphaGo** defeats Go champion Lee Sedol, showcasing AI's ability to master highly complex tasks.

2012 - **AlexNet** wins the ImageNet competition, demonstrating deep learning's dominance in image recognition tasks.

2006 - **Deep learning** resurfaces as a powerful technique with Geoffrey Hinton's breakthroughs in training deep neural networks.

1997 - IBM's **Deep Blue** defeats chess champion Garry Kasparov, showcasing AI's potential in strategic reasoning.

1990s - AI shifts toward probabilistic models; development of techniques like **Bayesian networks** for uncertainty handling.

1986 - **Backpropagation algorithm** revitalizes interest in neural networks, enabling multilayered learning models.

1980s - Rise of **Expert Systems**, AI programs designed to simulate decision-making of human experts in specific domains.

1970s - The **AI Winter** begins, as funding declines due to unmet expectations and limited computational power.

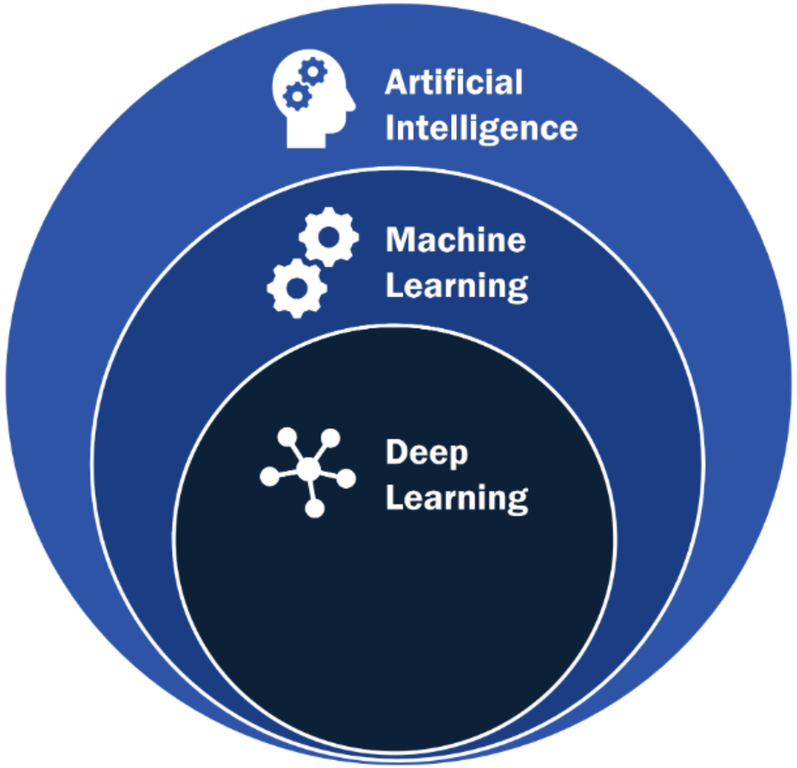
1960s - Early AI focuses on symbolic reasoning, with successes in problem-solving (e.g., Logic Theorist, General Problem Solver).

1956 - **Dartmouth Workshop**: The term "Artificial Intelligence" is coined, marking the official birth of AI as a field.

1943 - **McCulloch and Pitts** propose the first mathematical model of a neuron, laying the groundwork for neural networks.

What is Machine Learning and AI?

AI or ML or DL?



AI: systems that simulate human intelligence

- rule based systems
- expert systems
- robotics
- natural language processing
- computer vision
- machine learning

Today

ML: algorithms that learn and improve from data

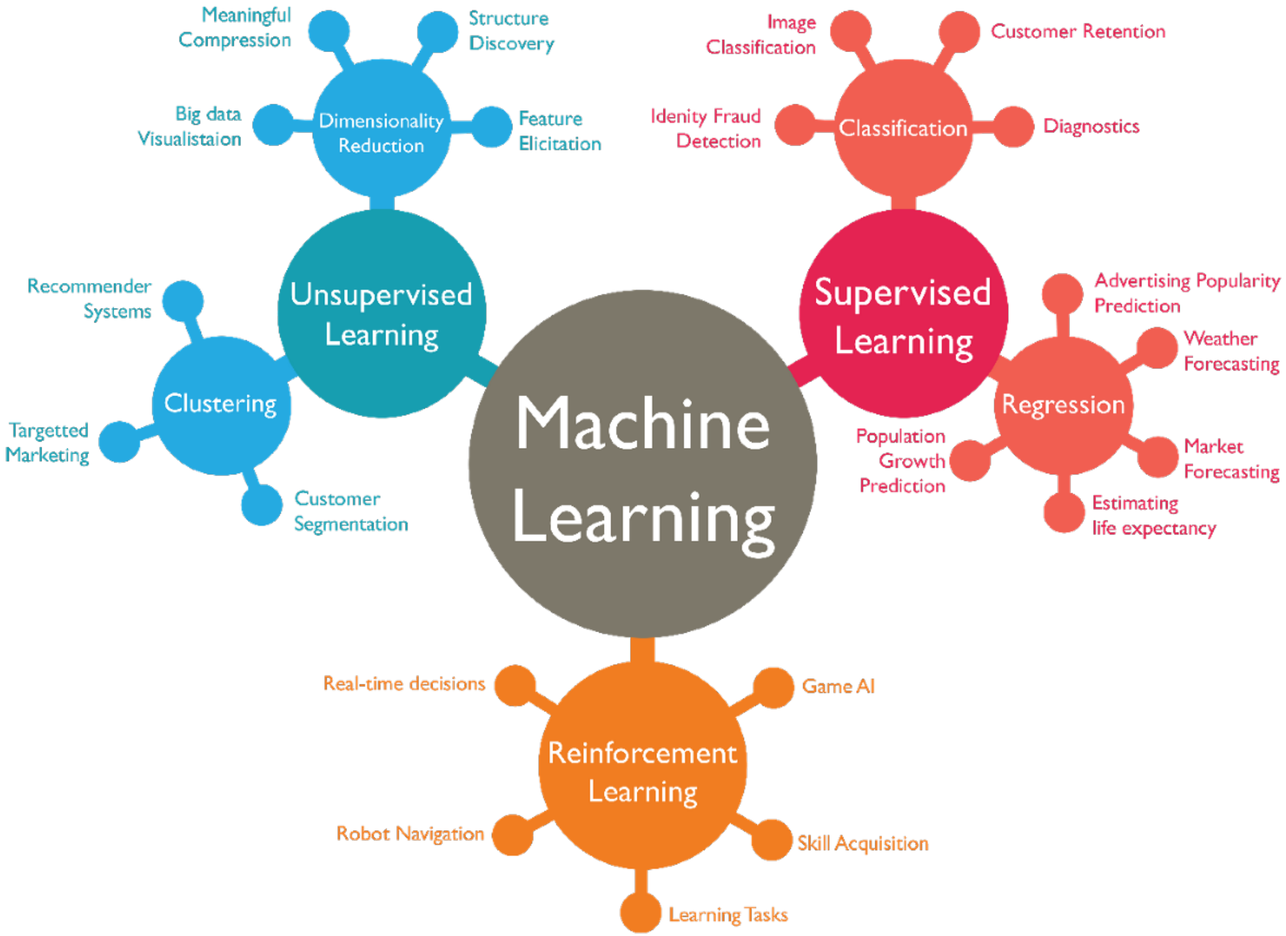
- supervised learning (regression & classification)
- unsupervised learning (clustering & dimensional reduction)
- reinforcement learning, Bayesian optimisation

DL: ML using large neural networks

- generative AI, LLMs, …

What is Machine Learning and AI?

Types of Machine Learning Tasks



1 Supervised learning

I have labeled data, (X, Y)

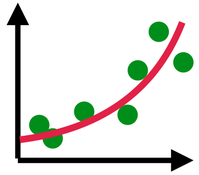
2 Unsupervised learning

My data has no labels, only (X)

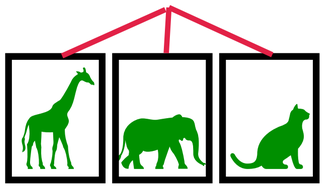
3 Reinforcement learning

I have no data yet; learning on-the-fly

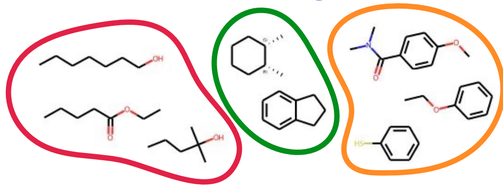
regression



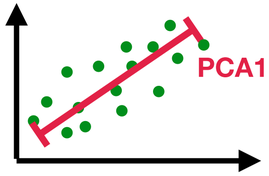
classification



clustering



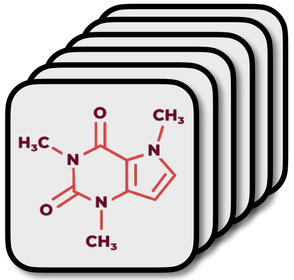
dimensional reduction

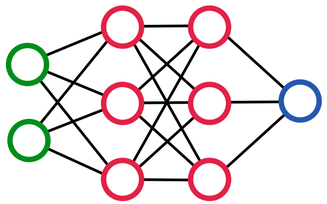


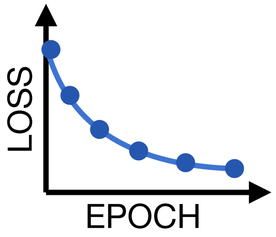
Bayesian optimization

Machine Learning Workflow

Step 0	1	2	3	4
Task Definition	Training Data	Choosing the Model	Training the Model	Inference







supervised learning *unsupervised learning*

- Which ML Task?: regression, classification, clustering, dimensional reduction, reinforcement learning
- ML Rule 1: do not start with machine learning! Try first with a (simple) base line approach.

A simple example

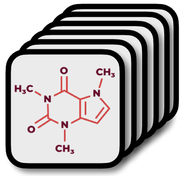
Chemical Structure – Property Prediction

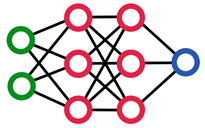
Given:

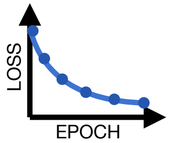
- **Task:**
  Predict solubility in water
- **Data:**
  1000 data points, list of molecules + their solubility in [mol/liter]

0  Task?

(Un-)Supervised / reinforcement ?
Regression / Classification?

1  Training data?

2  Model?

3 

Training?

4  Inference?

A simple example

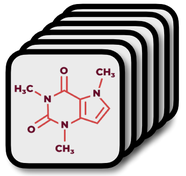
Chemical Structure – Property Prediction

Given:

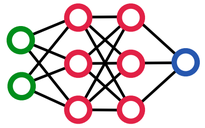
- **Task:**
  Predict solubility in water
- **Data:**
  1000 data points, list of molecules + their solubility in [mol/liter]

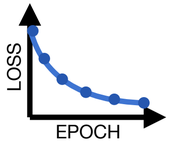
0  **Task:**

- Supervised Learning
- Regression

1  **Training data?**

- feature extraction
- EDA, exploratory data analysis
- molecular representations

2  **Model?**

3 

Training?

4  **Inference?**

Training Data

exploratory data analysis (EDA) and feature extraction

Exploratory data analysis

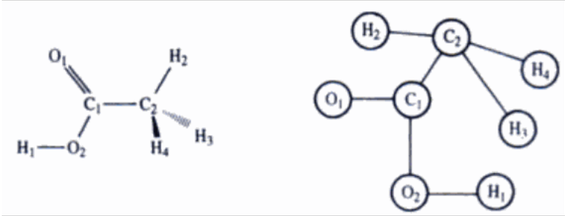
- examine first your training data sets
- compute some statistics (mean, variance)
- plot / visualize distributions
- How correlated are input features?
- remove outliers
- remove “nans” and “infs” etc.

Feature extraction

- use the raw data?
- pre-compute useful features
- scale, normalize input
- use molecular descriptors?
- learn / select feature with model?
- choice of input features is task dependent

Training Data

three molecular representations



Graphs

- nodes represent atoms, connections are bonds
- node and connection lists contain information, e.g. position, charge, mass, bond-order, etc.
- structure-property prediction with "graph neural networks (GNN)"

CC is CH3CH3 (ethane)
CC(=O)O is CH3COOH (acetic acid)

Strings, such as SMILES

- human readable, found in many databases
- easy to convert into graph or fingerprint
- syntax can be learned by language model
- not unique, e.g. CCO, OCC, C(O)C (=ethanol)
- syntax errors: CC(CCCC (missing closing parenthesis)
- semantic errors: CO=CC (unphysical oxygen with 3 bonds)

10010001000000101100000....

Fingerprints

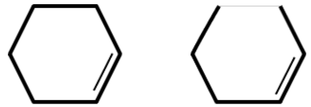
- long binary strings of "0" and "1"; typical length: 1024
- can encode molecular structure and topology, e.g. "has COOH", is aromatic", etc.
- or electronic properties, e.g. electronegativity, dipole moments, and partial charges
- or physico-chemical properties, solubility, lipophilicity,…
- good for property prediction and finding molecular similarity

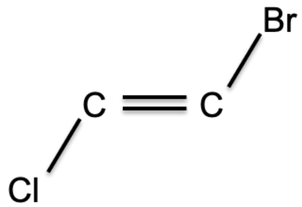
SMILES format

Basic guidelines:

- String of atom names + extra tokens
- B, C, N, O, F, P, S, Cl, Br, en I. Other atoms in square brackets: [Au].
- Hydrogens are implicit.
- Parentheses, (), indicate branches. E.g. CC(C)(C)C is dimethyl propane.
- Single bonds are implicit, = for double, # for triple
- Rings are noted broken, corresponding integers are added to the connecting atoms:
  Oc1ccccc1 is phenol (or e.g. C1=CC=C(C=C1)O)
- Double bond stereochemistry, use / and \ to denote cis and trans
  e.g. Cl/C=C/Br (trans), Cl/C=C\Br (cis)
- @ or @@ for tetrahedral stereochemistry:
  e.g. Br[C@](Cl)(I)F from the Br, the Cl, I, and F are arranged anticlockwise
- Aromaticity, use lower case
  e.g. C1CCCCC1 (cyclohexane)
  e.g. c1ccccc1 (benzene)

C1CCC=CC1





A simple example

Chemical Structure – Property Prediction

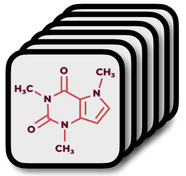
Given:

- **Task:**
  Predict solubility in water
- **Data:**
  1000 data points, list of molecules + their solubility in [mol/liter]

0 

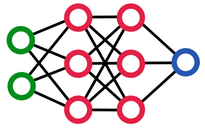
Task:

- Supervised Learning
- Regression

1 

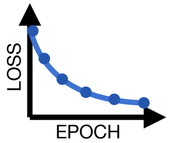
Training data:

- feature extraction
- molecular representations
- EDA, exploratory data analysis

2 

Model?

- Decision trees
- Neural Networks

3 

Training?

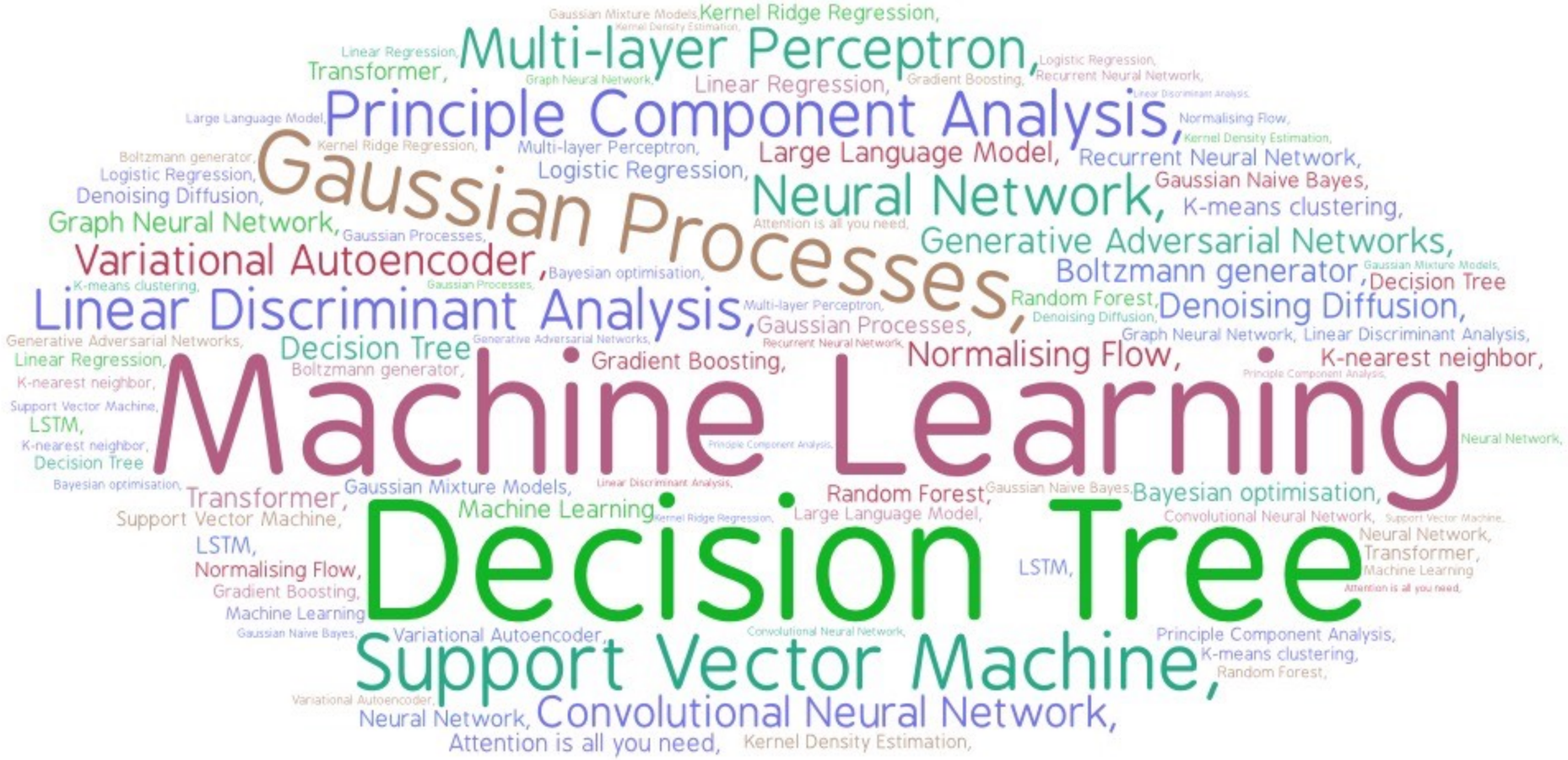
4 

Inference?

Choosing the model

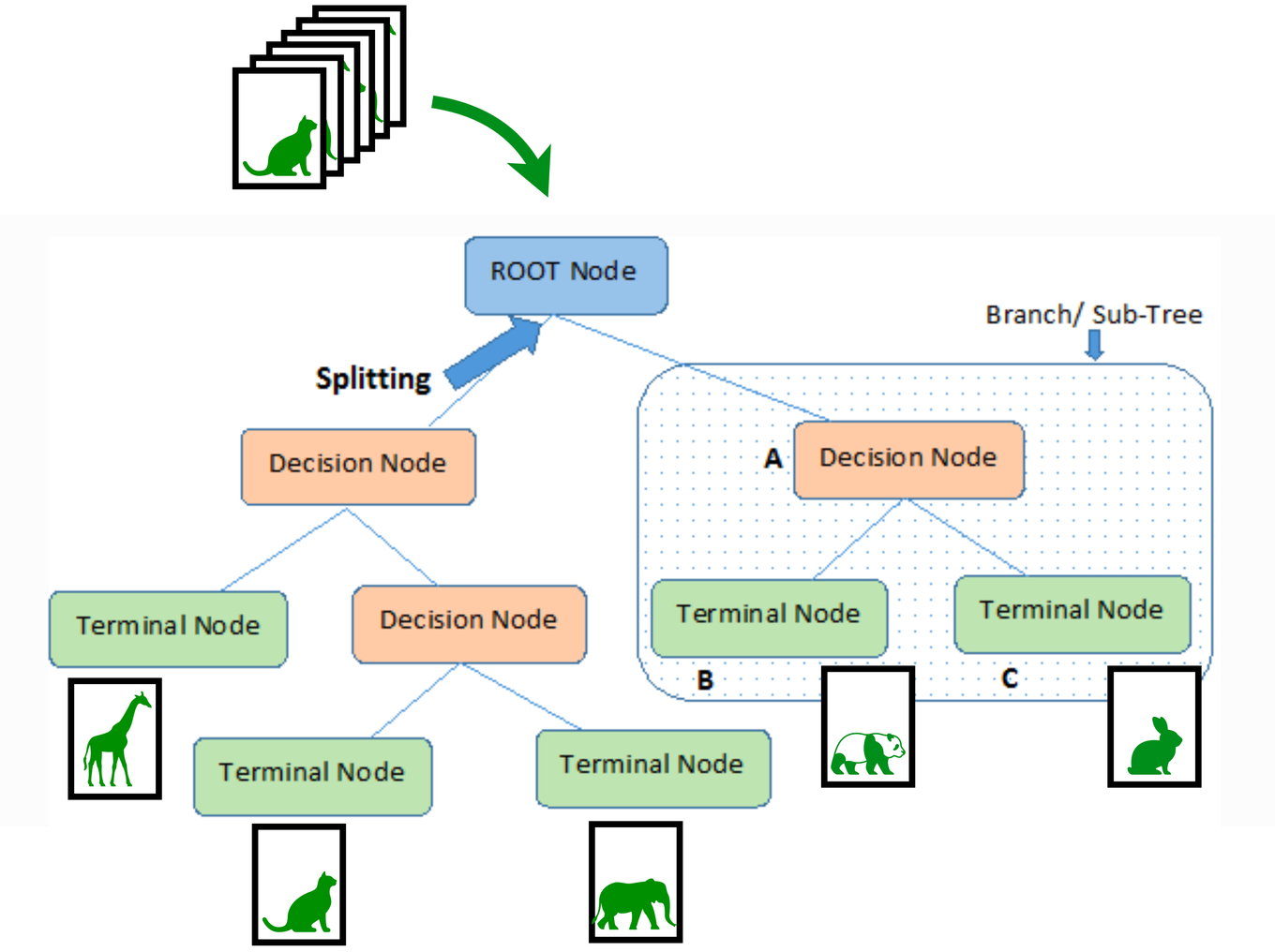
Depends on:

- the task
- problem complexity
- data
- stability
- transferability
- probabilistic output
- interpretability
- …



Choosing the model

Decision trees



Goal:

- Classification or Regression
- Learn decision rules from input features

Algorithm:

1. Start at root node, with entire dataset.
2. Find best separating feature (entropy reduction)
3. Generate the decision tree node with the optimally separating feature.
4. Recursively make new decision nodes (goto 2) until at terminating leaf node

Pros / cons:

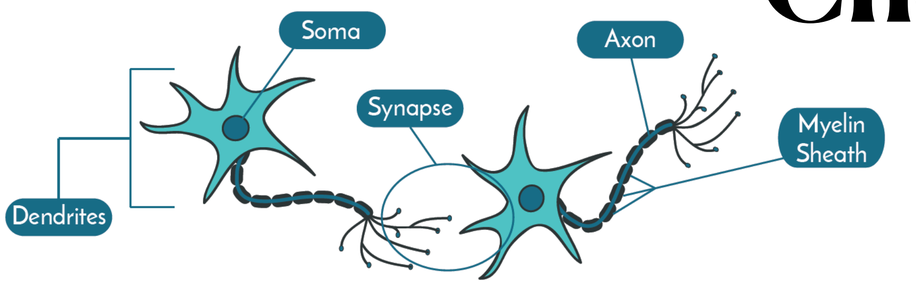
- Simple, white box (interpretable), not costly
- Pruning required to avoid overfitting

- advanced random forests & boosting models

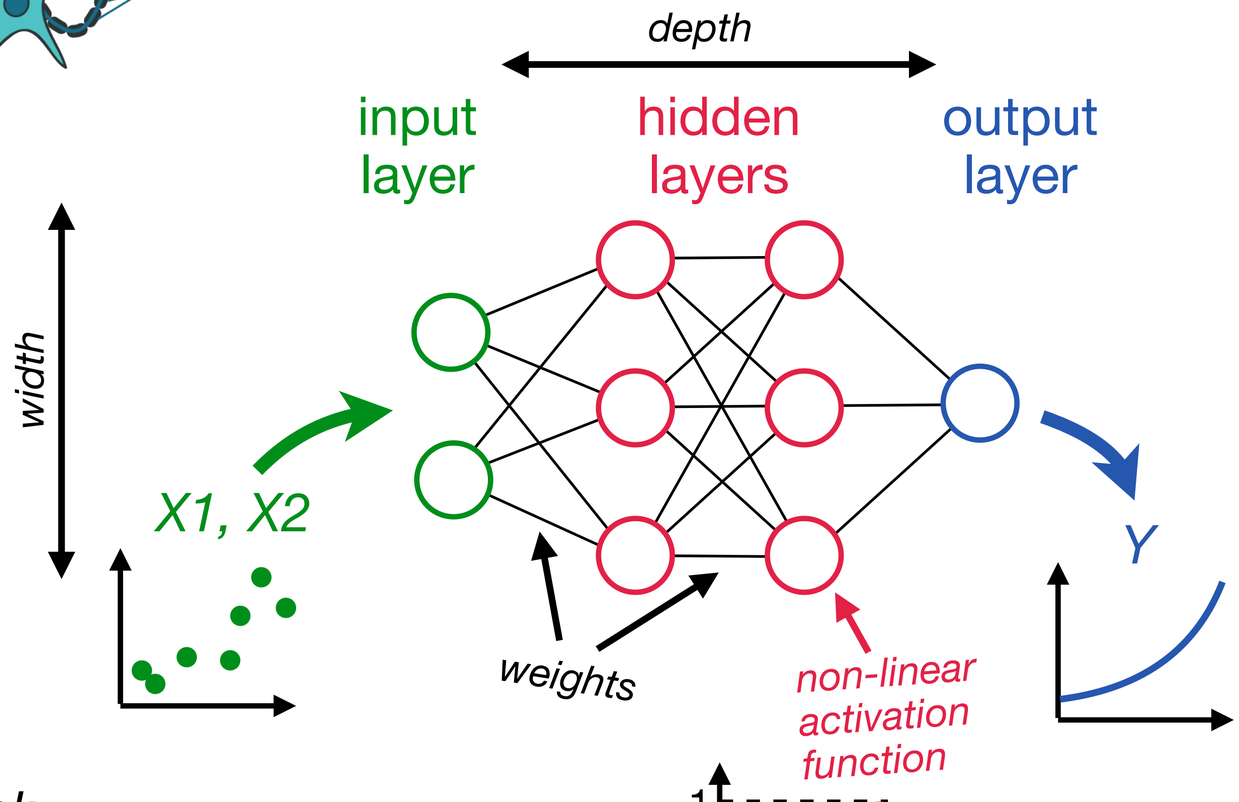
Choosing the model

Neural Networks

neurons in the brain



- ~ 86 10^9 neurons in human brain
- 10^3 -10^4 synapses per braincell
- 10^{14} - 10^{15} synapsen (connections) in human brain
- largest ML models: 10^{12} parameters



mathematical model:

$$y(\mathbf{x}, \mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{w}^{(3)}) = \sum_{j=0}^{J} w_j^{(3)} f_2\left(\sum_{i=0}^{I} w_{ji}^{(2)} f_1\left(\sum_{d=0}^{D} w_{id}^{(1)} x_d\right)\right)$$

Goal:

- Classification or Regression
- Learn weights of (hidden) function

Algorithm:

1. Forward propagation: compute output from input
2. Back propagation: apply chain-rule to get gradients
3. Optimize weights and goto 1. until convergence

Pros / Cons:

- “universal approximator”, many architectures possible
- interpretation more difficult

A simple example

Chemical Structure – Property Prediction

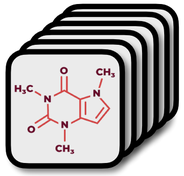
Given:

- **Task:**
  Predict solubility in water
- **Data:**
  1000 data points, list of molecules + their solubility in [mol/liter]

0 

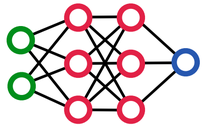
Task:

- Supervised Learning
- Regression

1 

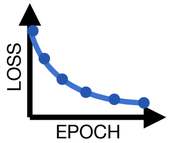
Training data:

- feature extraction
- molecular representations
- EDA, exploratory data analysis

2 

Model:

- Decision trees
- Neural Networks

3 

Training?

- Loss-function
- Training-Validation-Testing
- Regularisation

4 

Inference?

Training and Validation

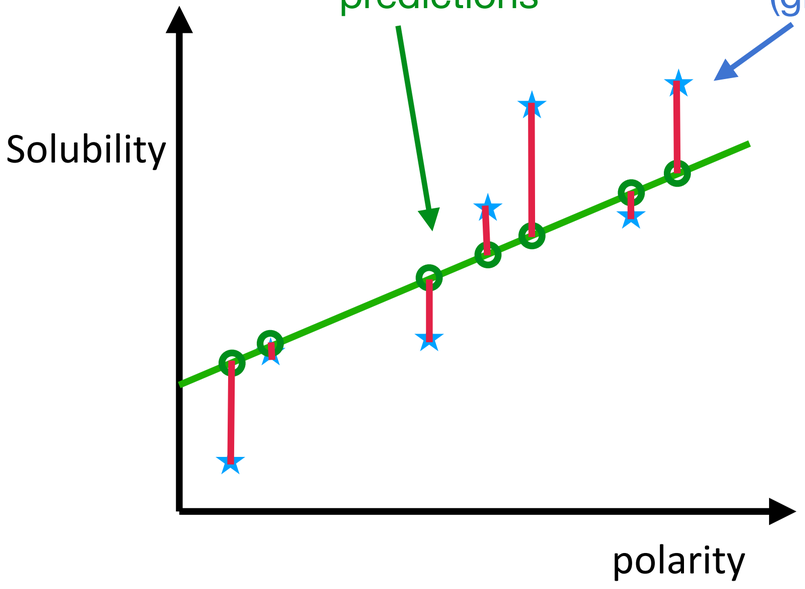
The Loss Function

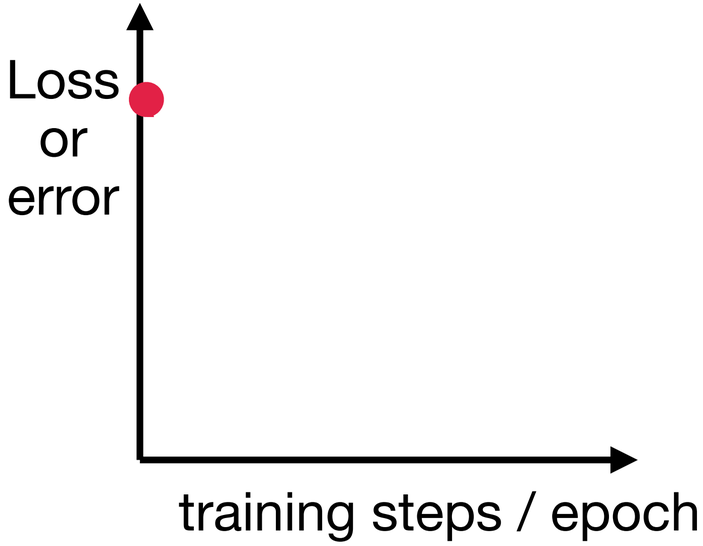
Loss (error) function:

$$E(\mathbf{w}) = \sqrt{\frac{1}{N_{\text{samples}}} \sum_{i=1}^{N_{\text{samples}}} \left(y(x_i, \mathbf{w}) - t_i\right)^2}$$

model predictions

measurements (ground truth)



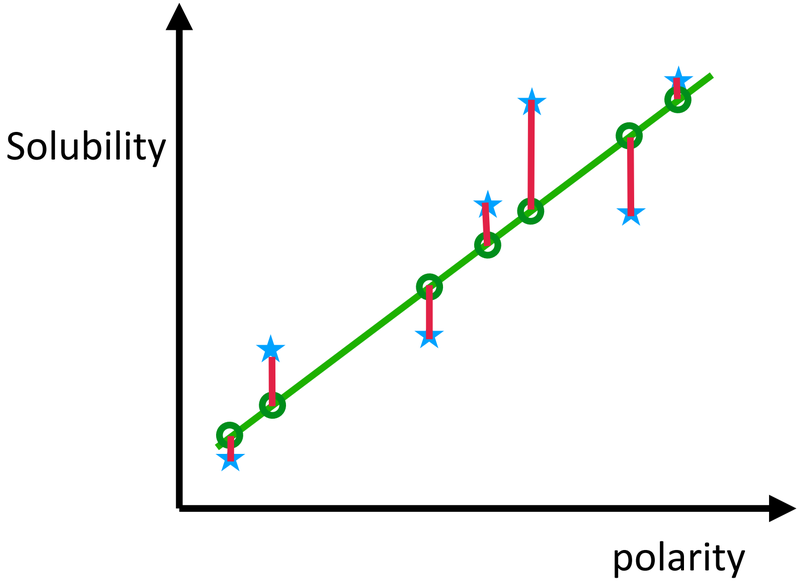


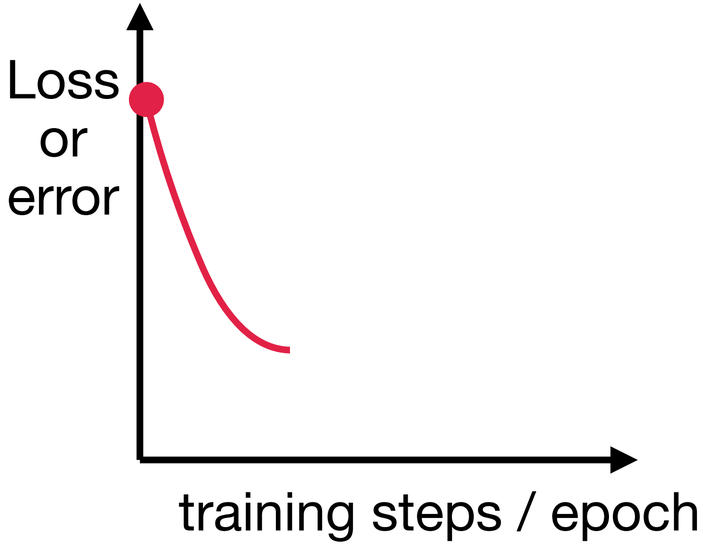
Training and Validation

The Loss Function

Loss (error) function:

$$E(\mathbf{w}) = \sqrt{\frac{1}{N_{\text{samples}}} \sum_{i=1}^{N_{\text{samples}}} \left(y(x_i, \mathbf{w}) - t_i\right)^2}$$



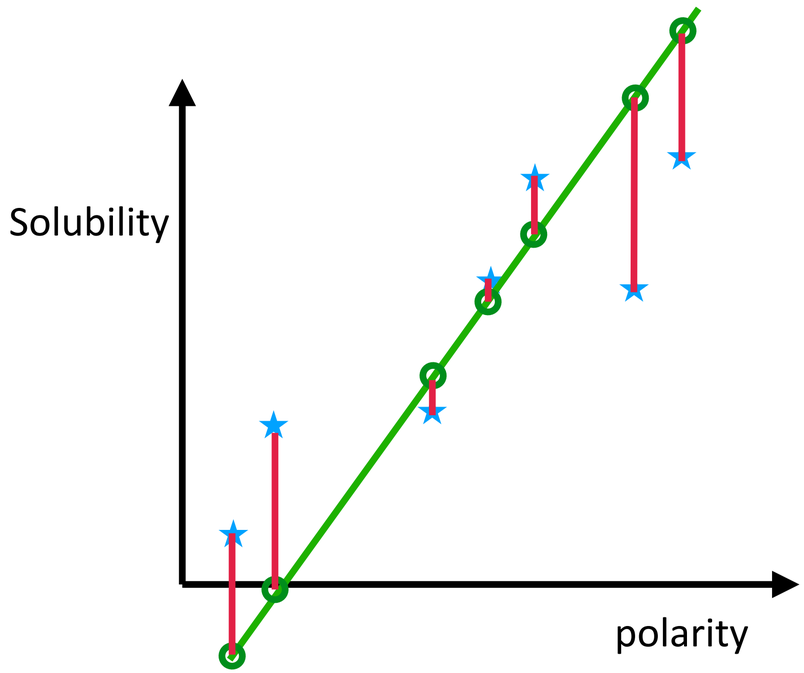


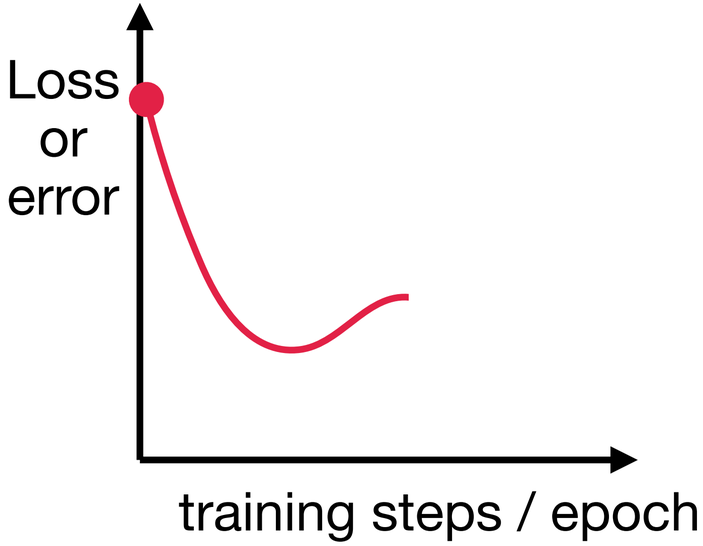
Training and Validation

The Loss Function

Loss (error) function:

$$E(\mathbf{w}) = \sqrt{\frac{1}{N_{\text{samples}}} \sum_{i=1}^{N_{\text{samples}}} \left(y(x_i, \mathbf{w}) - t_i\right)^2}$$



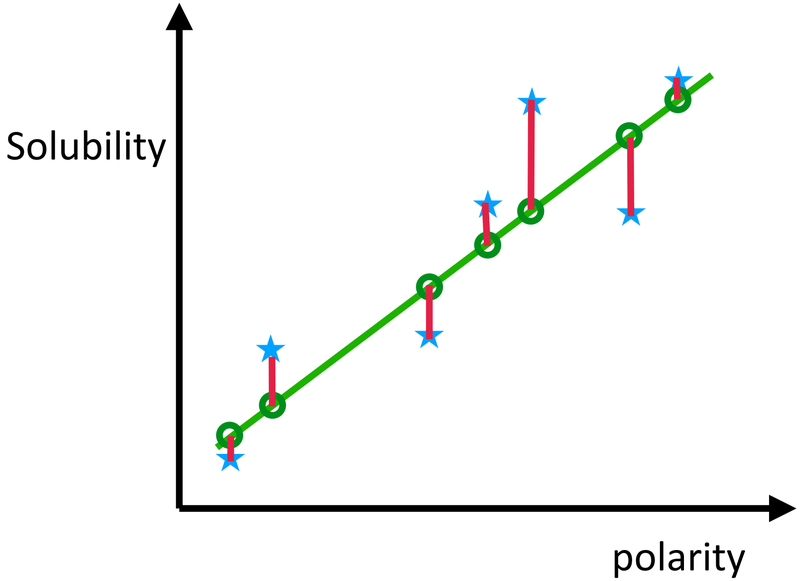


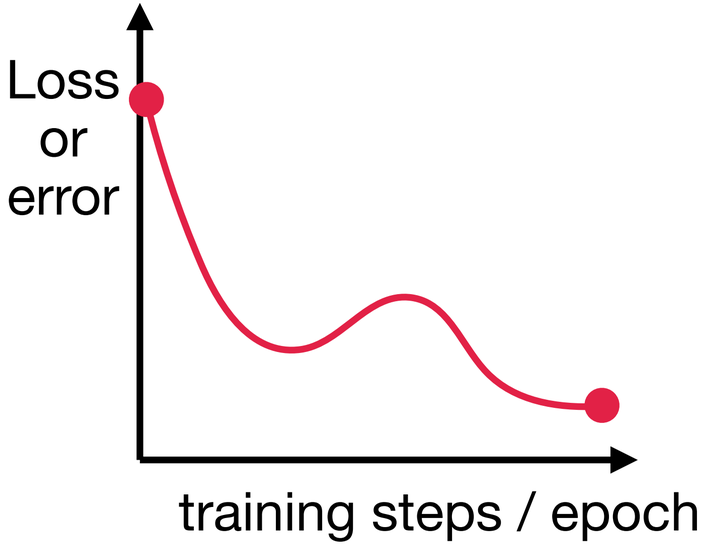
Training and Validation

The Loss Function

Loss (error) function:

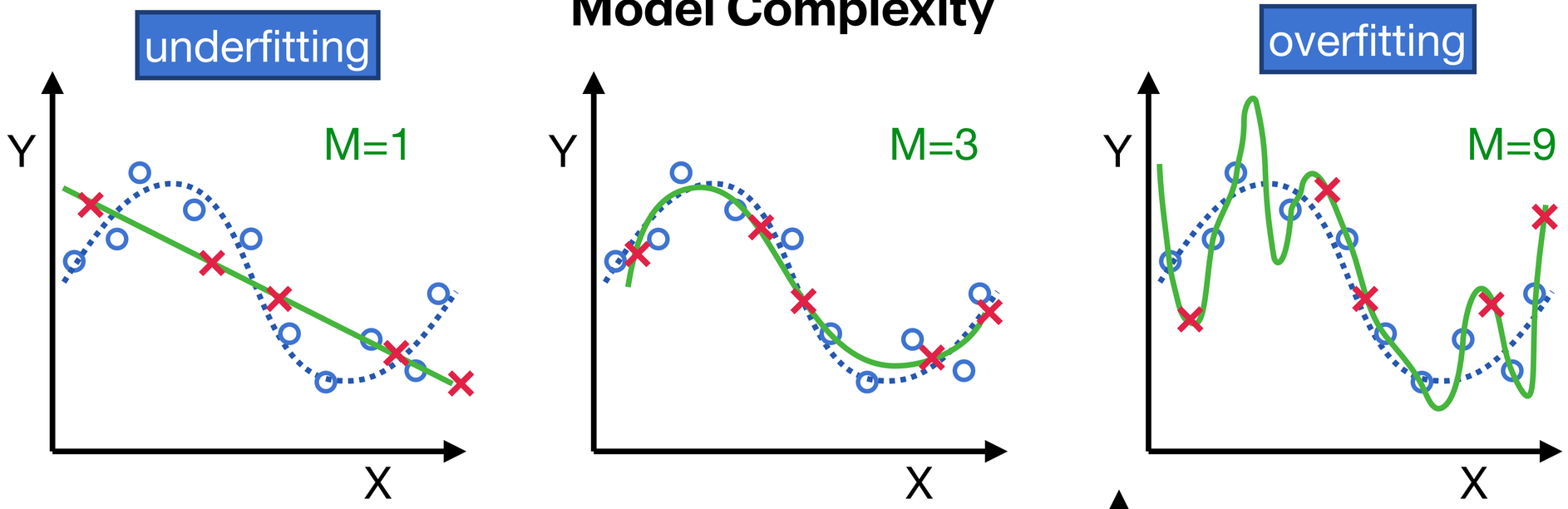
$$E(\mathbf{w}) = \sqrt{\frac{1}{N_{\text{samples}}} \sum_{i=1}^{N_{\text{samples}}} \left(y(x_i, \mathbf{w}) - t_i\right)^2}$$





Training and Validation

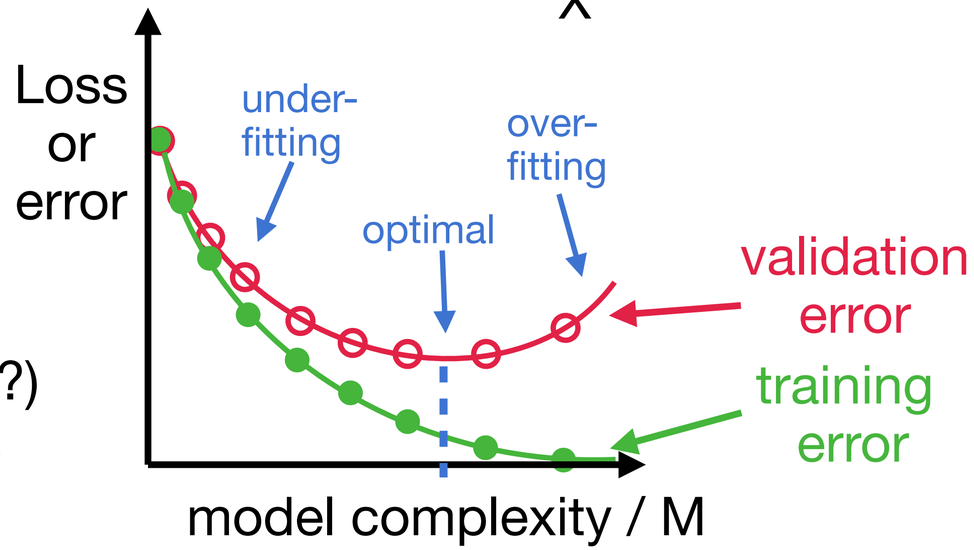
Model Complexity



Polynomial model function:

$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M = \sum_{i=0}^{M} w_i x^i$$

Does our model *generalise* to new data? (i.e. predictive power?)
-> Split data into training data and validation data *beforehand*.
-> Validation to test for under/over-fitting!



“With four parameters I can fit an elephant and with five I can make him wriggle his trunk”

– John von Neumann

1903 – 1957

Training and Validation

Bias - Variance trade-off and Regularisation

Does our model *generalise* to new data? (i.e. predictive power?)
-> Split data into training data and validation data *beforehand*.
-> Validation to test for under/over-fitting!

Generally in ML:
too many parameters!
(more than datapoints)
=> watch out for over-fitting!

Problem:

underfitting

(large bias)

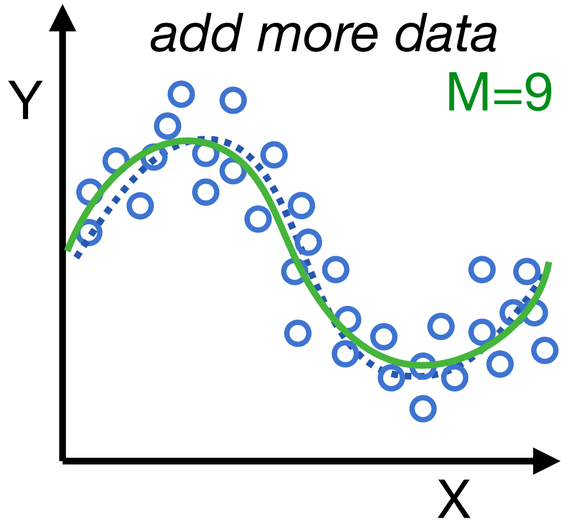
overfitting

(large variance)

Solutions:

- Increase model complexity (e.g. add more parameters)
- decrease regularisation

- Decrease model complexity
- Add more data
- Add some regularisation



Loss function with regularisation term: $L = \frac{1}{N}\sum_{i}^{N}\left[t_i - f(\mathbf{x}_i, \mathbf{w})\right]^2 + \lambda \sum_{k} w_k^2$

A simple example

Chemical Structure – Property Prediction

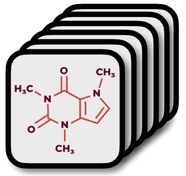
Given:

- **Task:**
  Predict solubility in water
- **Data:**
  1000 data points, list of molecules + their solubility in [mol/liter]

0 

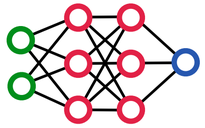
Task:

- Supervised Learning
- Regression

1 

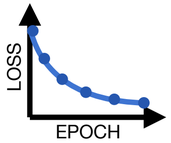
Training data:

- feature extraction
- molecular representations
- EDA, exploratory data analysis

2 

Model:

- Decision trees
- Neural Networks

3 

Training:

- Loss-function
- Training-Validation-Testing
- Regularisation

4 

Inference?

- Probabilistic/Bayesian approach
- Generative AI

Inference

Part II
Applied AI in Chemistry

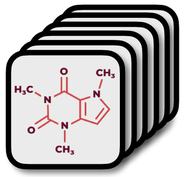
Summary

An Introduction to AI

0 

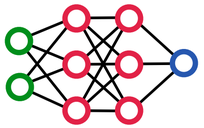
Task:

- Supervised Learning
- Regression

1 

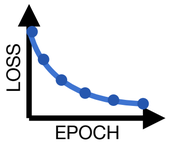
Training data:

- feature extraction
- molecular representations
- EDA, exploratory data analysis

2 

Model:

- Decision trees
- Neural Networks

3 

Training:

- Loss-function
- Training-Validation-Testing
- Regularisation

4 

Inference:

- Probabilistic/Bayesian approach
- Generative AI

Part I
An Introduction to AI
15:00 – 15.40

Part II
Applied AI in Chemistry
15:40 – 16.20

Part III
Exam / Quiz
16:20 – 17.00