Introduction to Machine Learning

Adeniyi Mosaku

Introduction to ML for Climate Scientists, DKRZ 04.03.24

Helmholtz AI

Artificial Intelligence Cooperation Unit

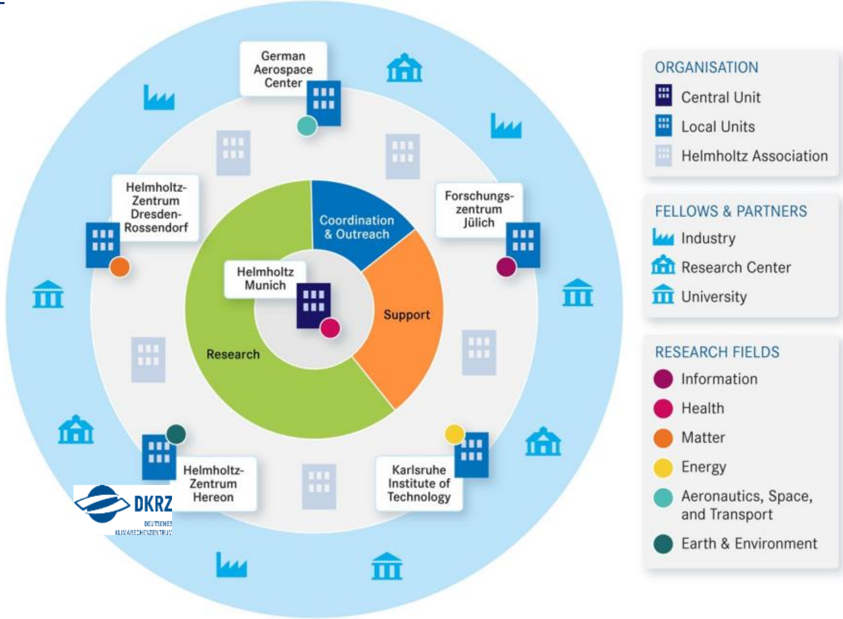
Mission
Bring applied AI / ML techniques to your research questions and datasets

6 LOCAL UNITS

Earth @ HZG
Energy @ KIT
Matter @ HZDR
Space @ DLR
Health @ HMGU
Info @ FZJ

Each Unit:

- Young Investigator Group
- AI Consultants

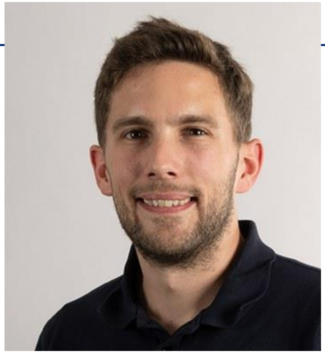


AI Consultants for Earth & Environment

DKRZ
DEUTSCHES KLIMARECHENZENTRUM

AWI
GEOMAR
UFZ HELMHOLTZ Zentrum für Umweltforschung
Helmholtz-Zentrum hereon
GFZ Helmholtz-Zentrum Potsdam
DLR Deutsches Zentrum für Luft- und Raumfahrt
JÜLICH Forschungszentrum
KIT

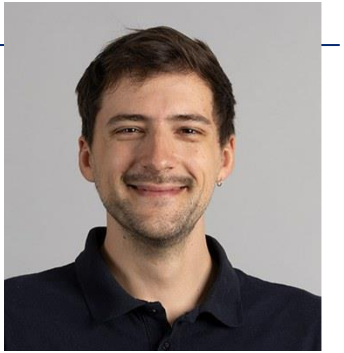
Christopher
Kadow



Johannes
Meuer

Étienne
Plésiat

Danai
Filippou



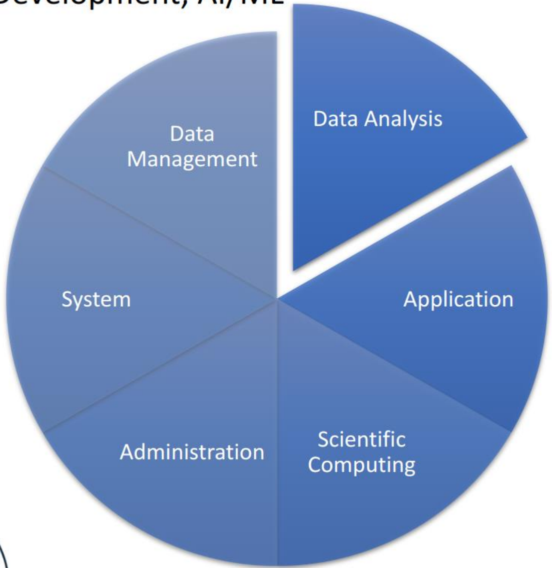
Max
Witte

CLINT group

DATA ANALYSIS DEPARTMENT

New Department: *Data Analysis*

Data Analysis, Mining and Handling, Scientific Software Development, AI/ML



Access to Petabytes of Climate Model Data

- Best evaluation of simulation

Access to Compute of the HPC including GPUs

- Acceleration of training of AI/ML

What is your name?

What do you expect to learn in this course?

Name one thing that you associate with machine learning

Introduction to Machine Learning for Climate Scientists

Workshop Outline

Day 1, March 4	
13:00	**Introduction to Machine Learning** (Adeniyi) • A comprehensive overview of the concepts and principle behind Machine Learning • Exploration of real-world applications of Machine Learning • Differentiating different types of Machine Learning • Introducing popular Machine Learning tools and frameworks
14:40	**Coffee Break**
15:00	**Architectures and Applications** (Paul) • An overview of state of the art Machine Learning Methods • Examples from weather, climate and beyond
16:30	**Explainable AI** (Harsh) • Introduction to Explainable AI (XAI) • Importance of Explainability • Interpretability techniques and use cases

Introduction to Machine Learning for Climate Scientists

Workshop Outline

Day 2, March 5	
9:00	**PyTorch: Application to Climate Science** (Etienne) ● Setup of the accounts ● Introduction to PyTorch with examples ● Definition of the task ● Creation of the training, validation and test datasets
10:30	**Coffee Break**
10:45	**PyTorch: Application to Climate Science** (Etienne) ● Building the CNN ● Training the model ● Testing the model
12:00	**Lunch Break**
13:30	**Advanced ML use-case: Reconstructing missing climate data** (Johannes) ● Create and modify inpainting CNN for reconstructing climate data ● Train the model with different configurations ● Validate the model on test data
15:30	**Closing Remarks**

DKRZ DEUTSCHES KLIMARECHENZENTRUM

Safety and Convenience

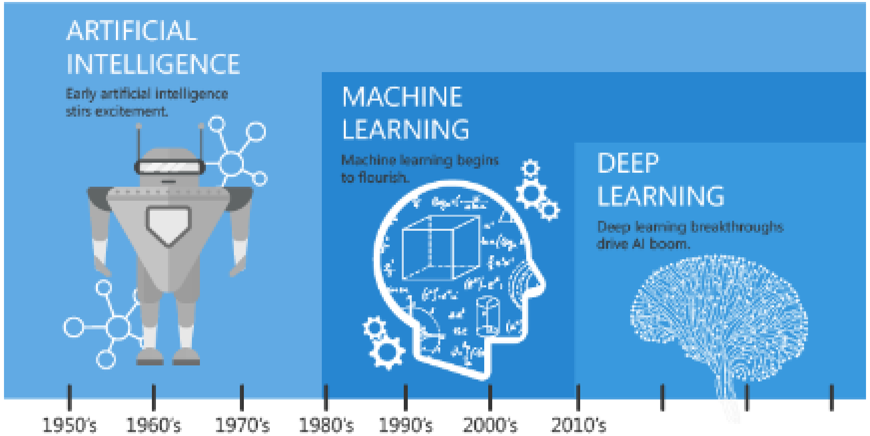
- Workshop WIFI:
  - SSID: `MLCS Workshop`
  - WPA2-PSK: `MLCS2024`

Coffee breaks are kindly sponsored by Helmholtz AI ☺

HELMHOLTZ AI

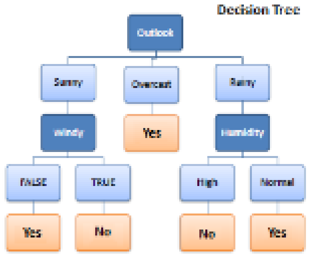
What is Machine Learning?

DKRZ
DEUTSCHES KLIMARECHENZENTRUM



Since an early flush of optimism in the 1950's, smaller subsets of artificial intelligence - first machine learning, then deep learning, a subset of machine learning - have created ever larger disruptions.

- Machine learning algorithms build a model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so.



- Deep learning: uses *neural networks* as models

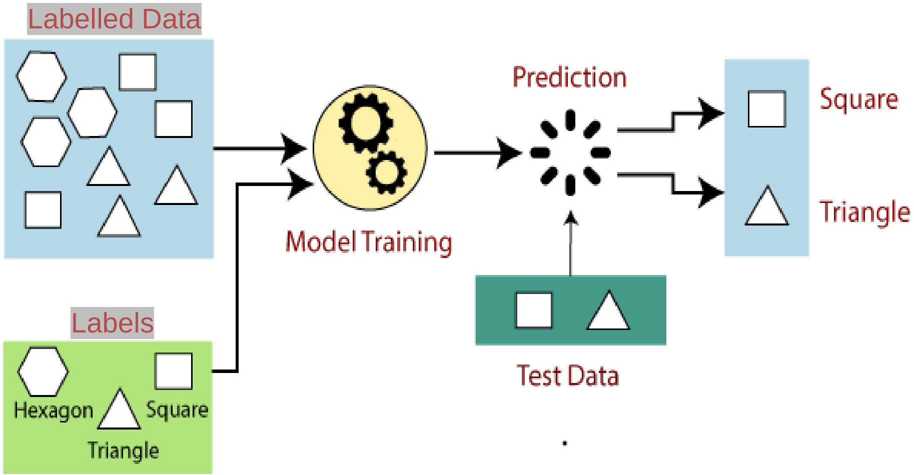
Machine Learning Progress

- **1990s**: Support Vector Machine
- **2010**: Deep Learning Resurgence
- **2014**: Generative Adversarial Networks
- **2015**: AlphaGo
- **2016**: AlphaGo Zero
- **2018**: Transformer (BERT)
- **2020**: AlphaFold
- **2020**: GPT-3 (Generative Pre-trained Transformer 3)

Types of Machine Learning

Supervised Machine Learning

- The algorithm is trained on a labelled dataset
- Input data is paired with corresponding target labels.

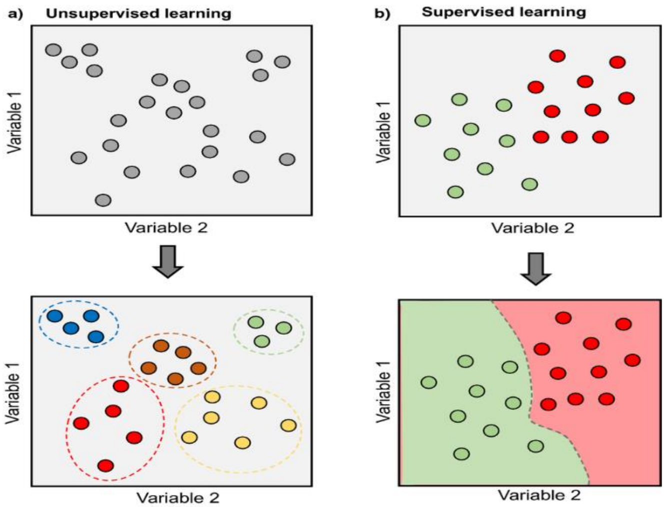


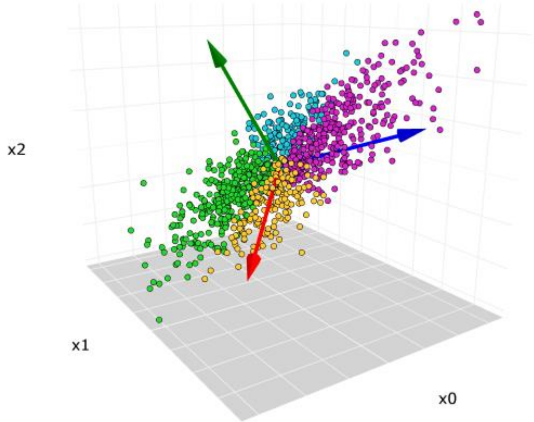
- Example: Classification, Regression

Types of Machine Learning

Unsupervised Machine Learning

- The algorithm is trained on an unlabelled dataset
- Discover hidden patterns, relationships, or clusters within the data.



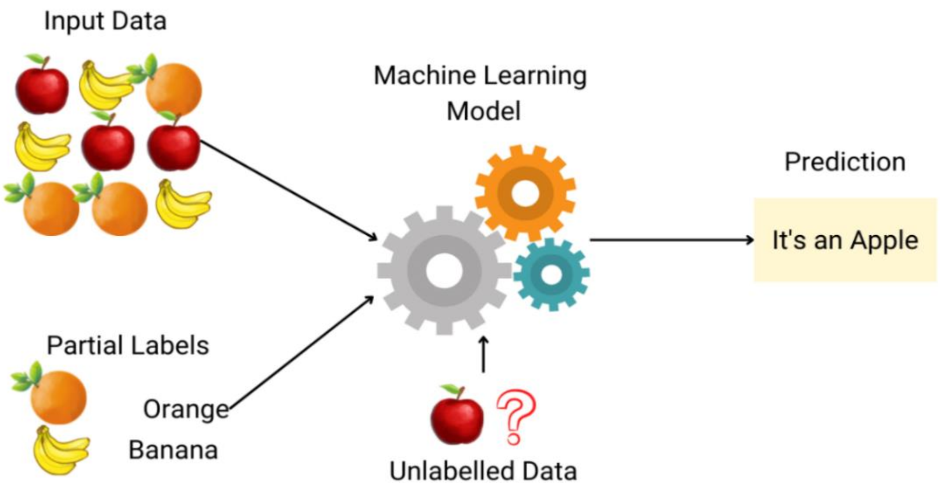


- Example: Clustering, Dimensionality Reduction, PCA

Types of Machine Learning

Semi-supervised Machine Learning

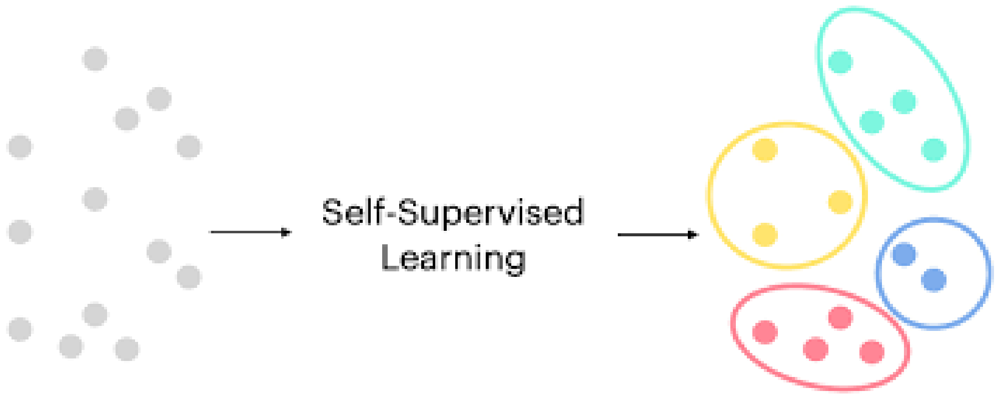
- The algorithm is trained on an labelled and unlabelled dataset
- Leveraging on labelled and unlabelled data to improve performance
- It saves time from data labelling



Types of Machine Learning

Self-supervised Machine Learning

- A special type of unsupervised learning
- The algorithm generates its own labels from input dataset
- It does not require external labels

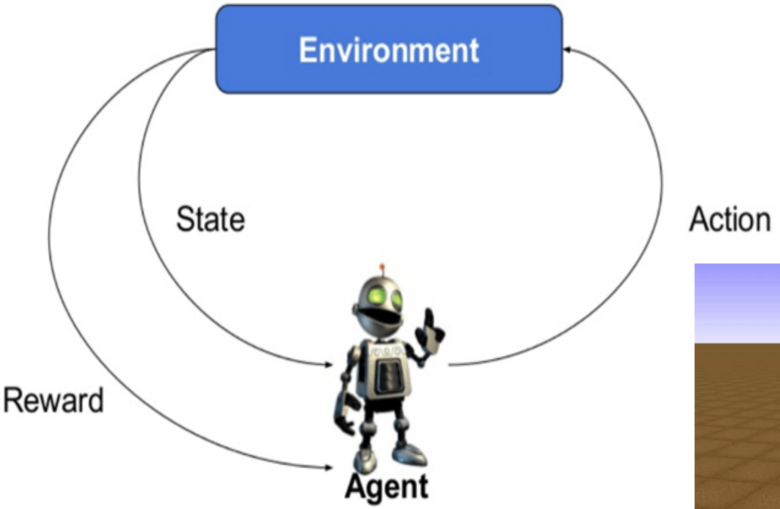


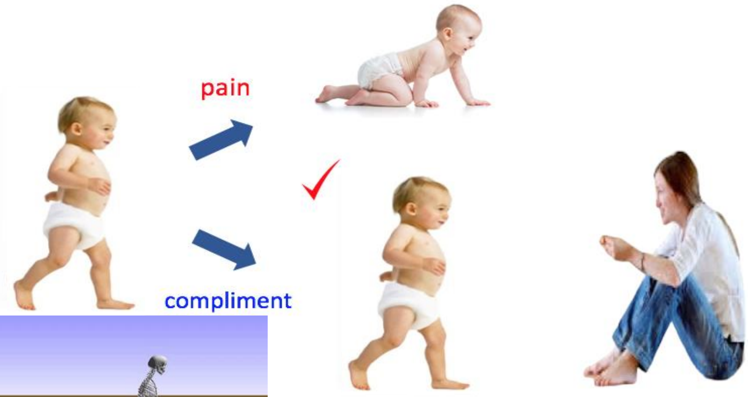
- Example: Auto-encoders, Contrastive learning

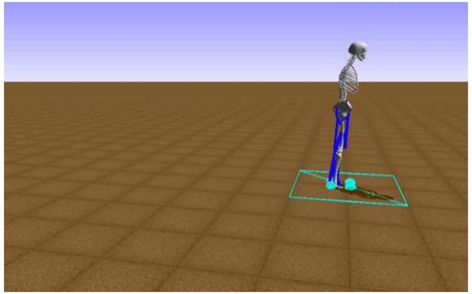
Types of Machine Learning

Reinforcement Machine Learning

- An agent interacting with an environment and learning based on feedback
- Learn a policy that maximizes cumulative reward over time





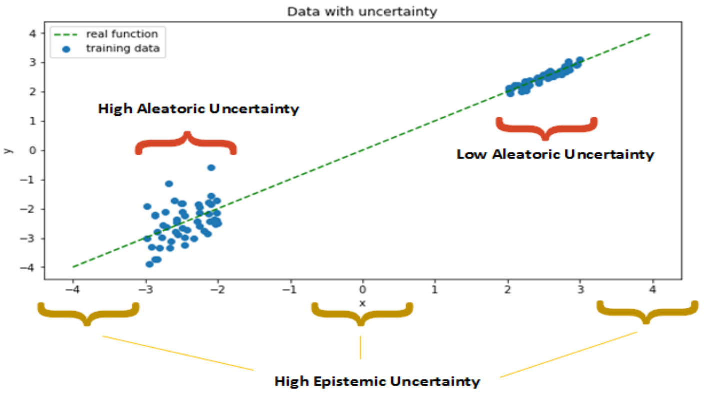


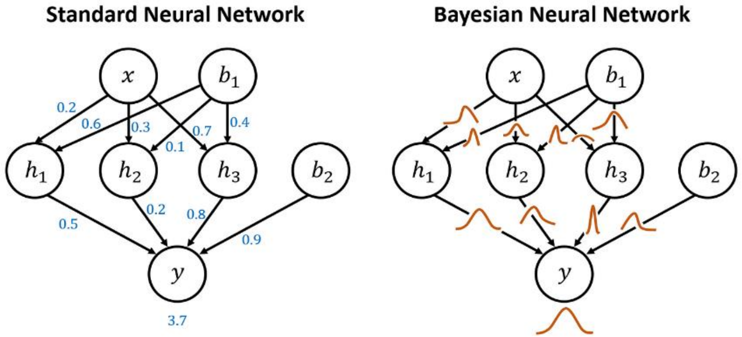
- Example: Humanoid robots, Games, autonomous system

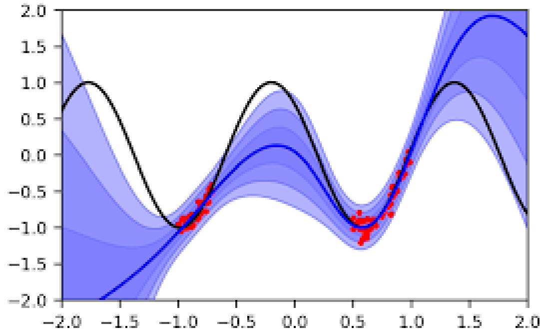
Types of Machine Learning

Probabilistic Machine Learning

- Data and Model include uncertainties
- We need to capture these uncertainties
- Probability distributions are maintained over weight



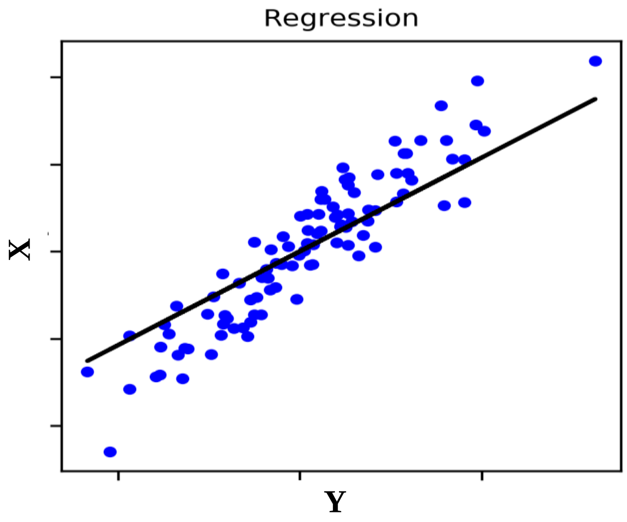




- Example: Bayesian Neural Network BNN

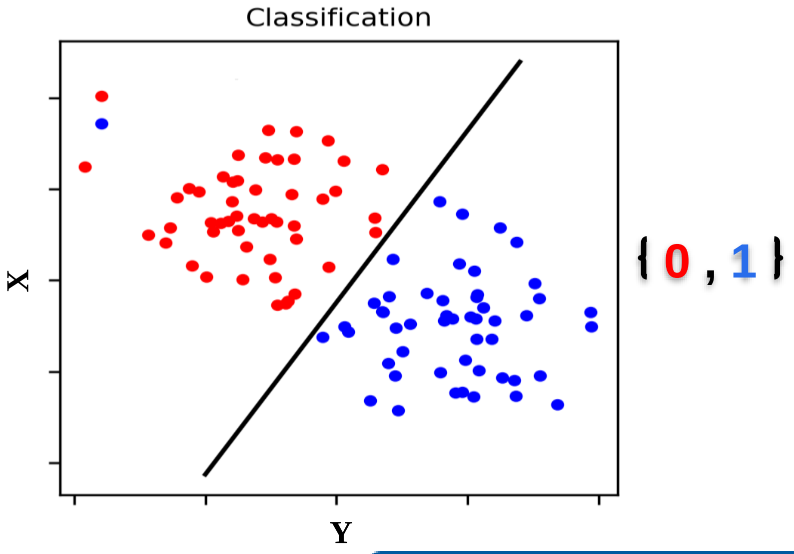
Regression and Classification in Machine Learning

Introduction



- Regression: Predicting a continuous outcome
- The output are continuous values

- Classification: Assigning instances to classes
- The output are probabilities using softmax



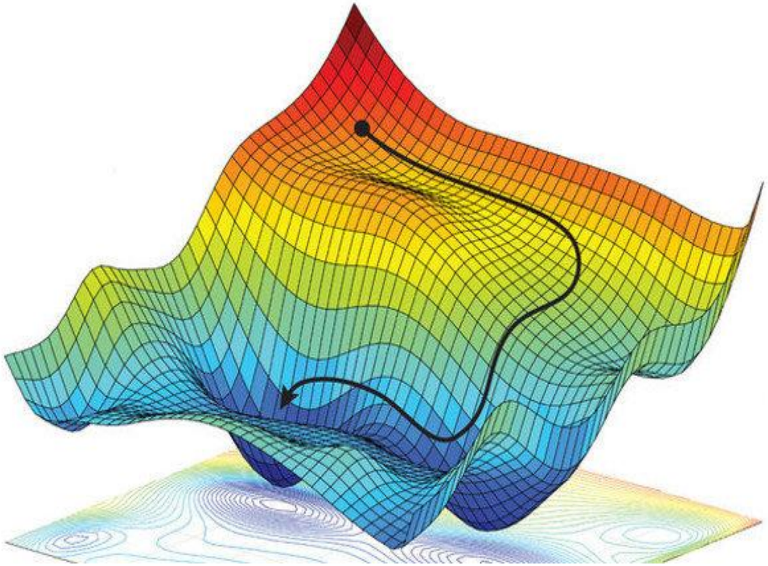
Regression and Classification in Machine Learning

Loss Function and Optimization

- Loss functions measure the disparity between predicted and actual values

- Aim is to minimize the loss function model

- Optimization helps to find the right and fastest path
  - Stochastic Gradient Descent (SGD)
  - Adaptive Moment (Adam)
  - Adaptive Gradient (Adagrad)
  - …



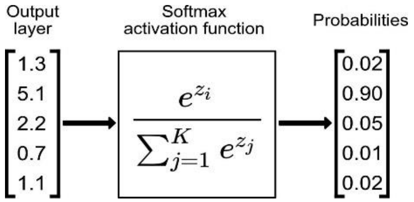
Regression and Classification in Machine Learning

Loss Function Types

- Regression: Mean Squared Error (MSE)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- Classification: Cross-Entropy Loss



$$\text{Cross Entropy Loss}_b = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$$

$$\text{Cross Entropy Loss}_m = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m} y_{ij}\log(\hat{y}_{ij})$$

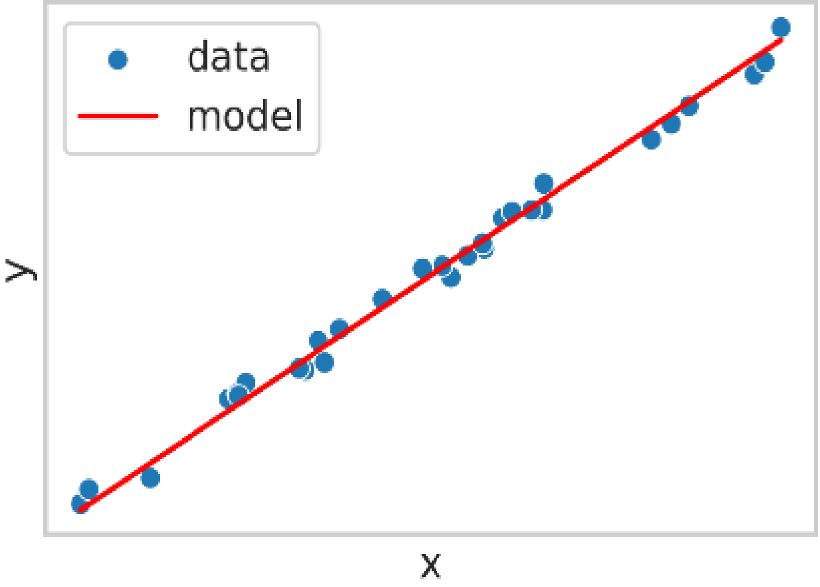
- Domain informed loss function
  - Created by domain scientists based on governing rules
  - Example: PINN (Physics Informed Neural Network)

Typical Machine Learning Procedure

Simple Linear Regression

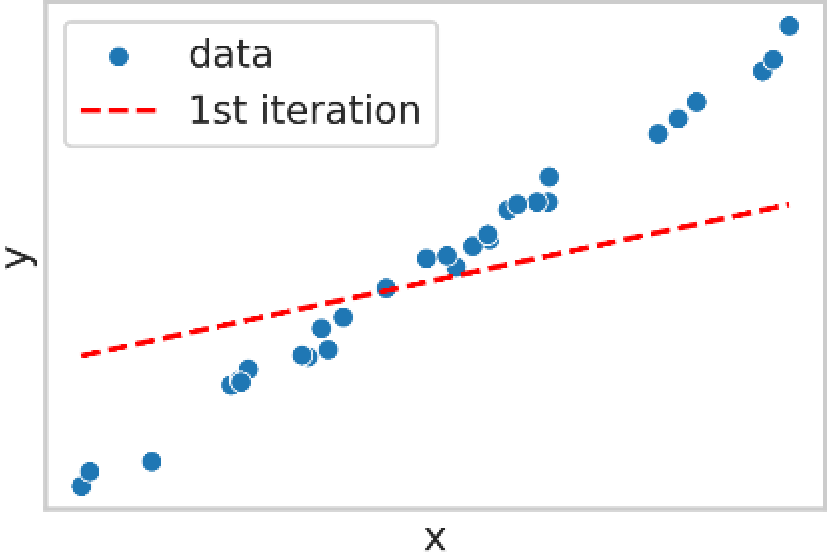
Are these data labeled or not?

- Data set
  - $D = \{features, labels\} = \{x, y\}$
- Model
  - Defined as $\hat{y} = wx + b$
  - Trainable parameters w, b
- Loss function
  - $L(w, b) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y_i})^2 = 1$
- Training: minimize the loss function
  - → parameters $\hat{w}, \hat{b}$



Slide adapted from M. Götz, KIT

Optimizing by Gradient Descent

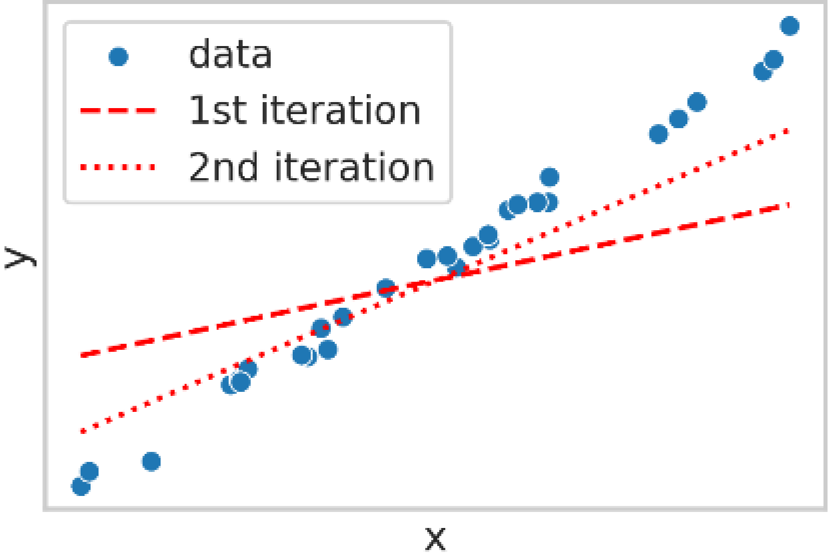


- Start with a random guess for the trainable parameters: w_i
- Calculate the loss function $L(w_i)$
- Parameter update in the direction of negative gradient

$$w_{i+1} = \mathrm{w_i} - \alpha \nabla_{w_i} L(w_i)$$

Slide adapted from M. Götz, KIT

Optimizing by Gradient Descent



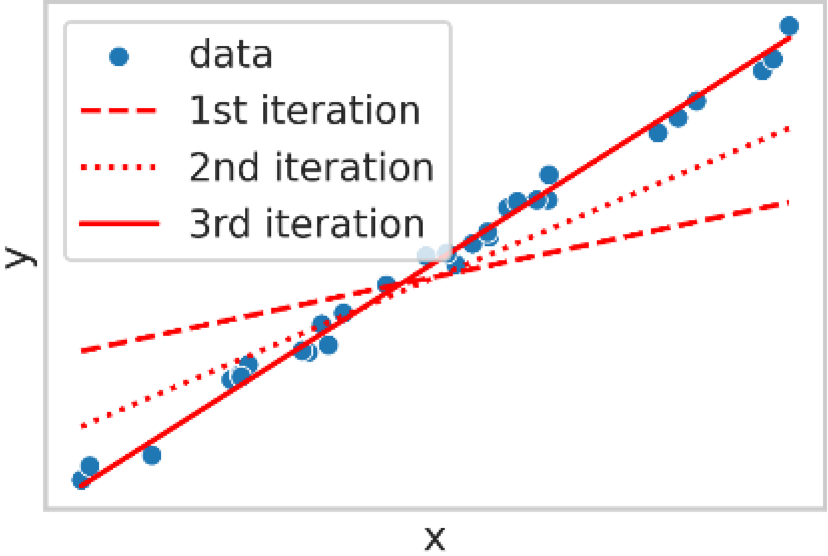
- Start with a random guess for the trainable parameters: w_i
- Calculate the loss function $L(w_i)$
- Parameter update in the direction of negative gradient

$$w_{i+1} = \mathrm{w_i} - \alpha \nabla_{w_i} L(w_i)$$

- Learning rate α (typically $\in [0.0001, 0.1]$)

Slide adapted from M. Götz, KIT

Optimizing by Gradient Descent



- Start with a random guess for the trainable parameters: w_i
- Calculate the loss function $L(w_i)$
- Parameter update in the direction of negative gradient

$$w_{i+1} = \mathrm{w_i} - \alpha \nabla_{w_i} L(w_i)$$

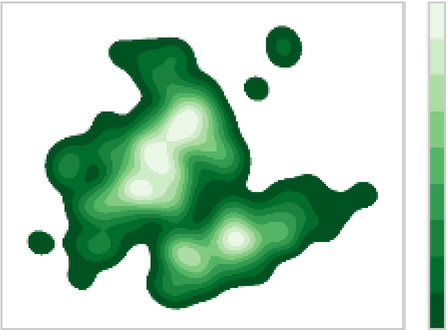
- Learning rate α (typically $\in [0.0001, 0.1]$)

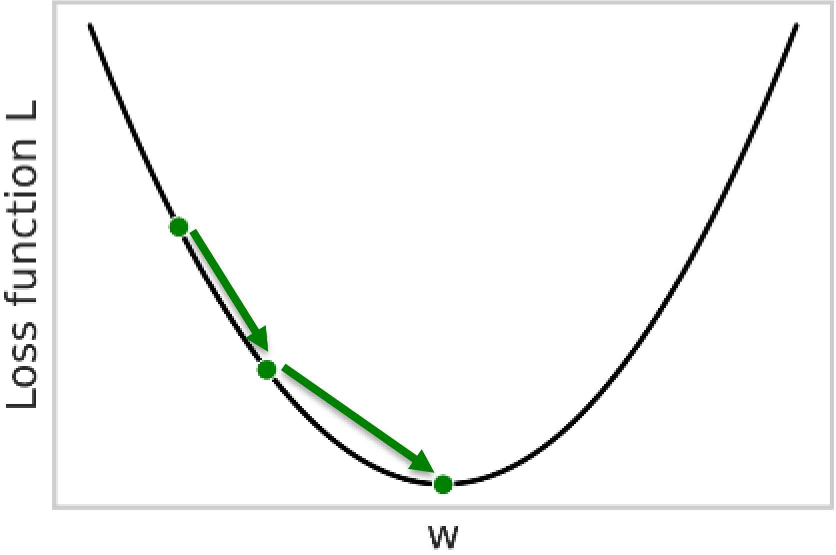
Visualizing the training procedure

- Weight update

$$w_{i+1} = \mathrm{w_i} - \alpha \nabla_{w_i} L(w_i)$$

- In practice: more than one trainable parameter → find local minimum

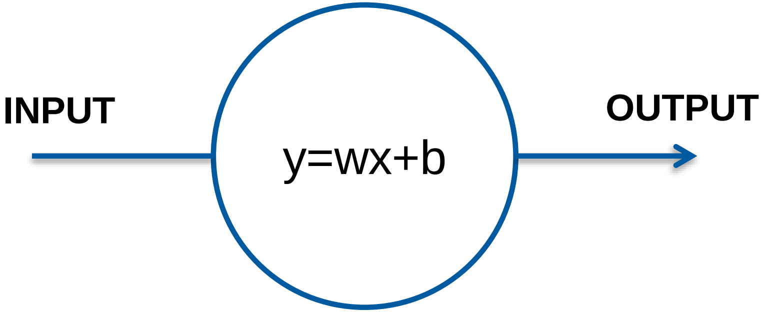




Slide adapted from M. Götz, KIT

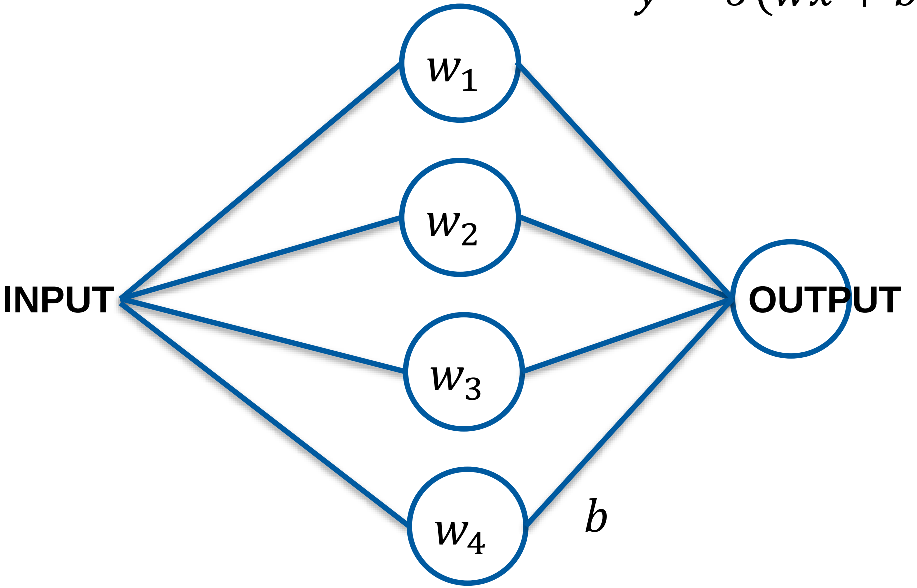
From linear regression to neural networks

- Linear regression: one „neuron“



- Neural network: stack neurons and add nonlinear activation function

$$\hat{y} = \sigma(wx + b)$$



Neural network

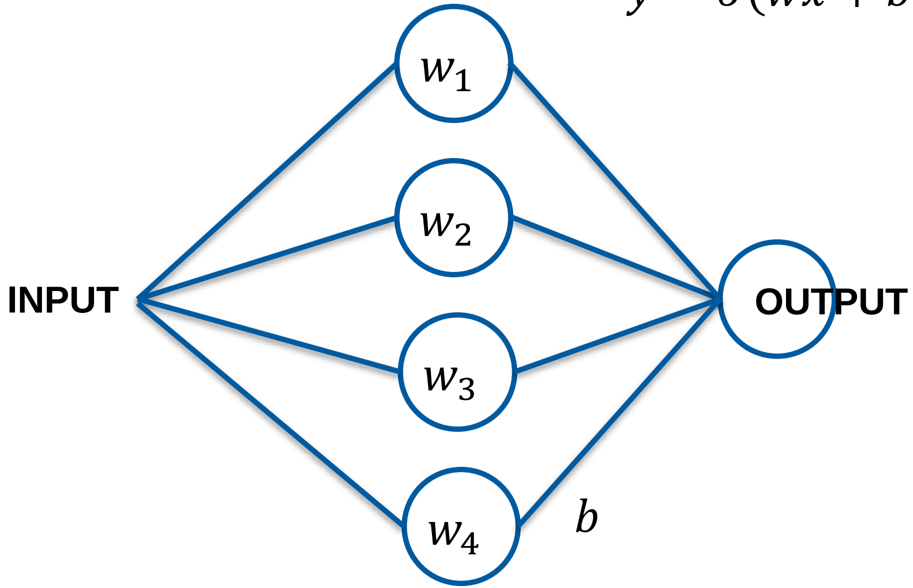
- Universal approximation theorem: NN can approximate any „well-behaved“ non-linear function
- Now: 5 trainable parameters

$$L = L(w_1, w_2, w_3, w_4, b)$$

- Large language models: 175 billion trainable parameters

- Neural network: stack neurons and add nonlinear activation function

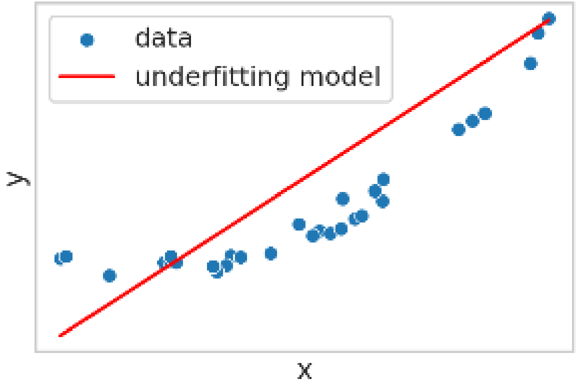
$$\hat{y} = \sigma(wx + b)$$

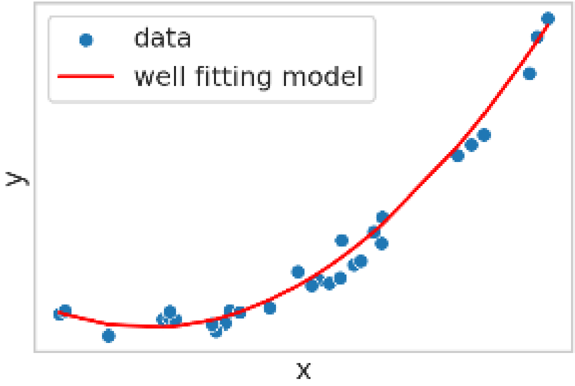


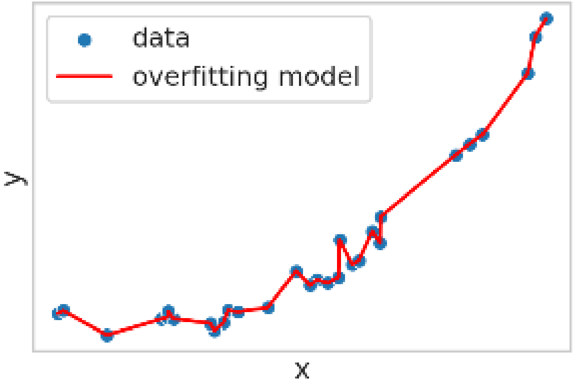
Evaluating Machine Learning Algorithms

Overfitting

- Overfitting: neural network learns to reproduce training data exactly
  → Does not generalize well



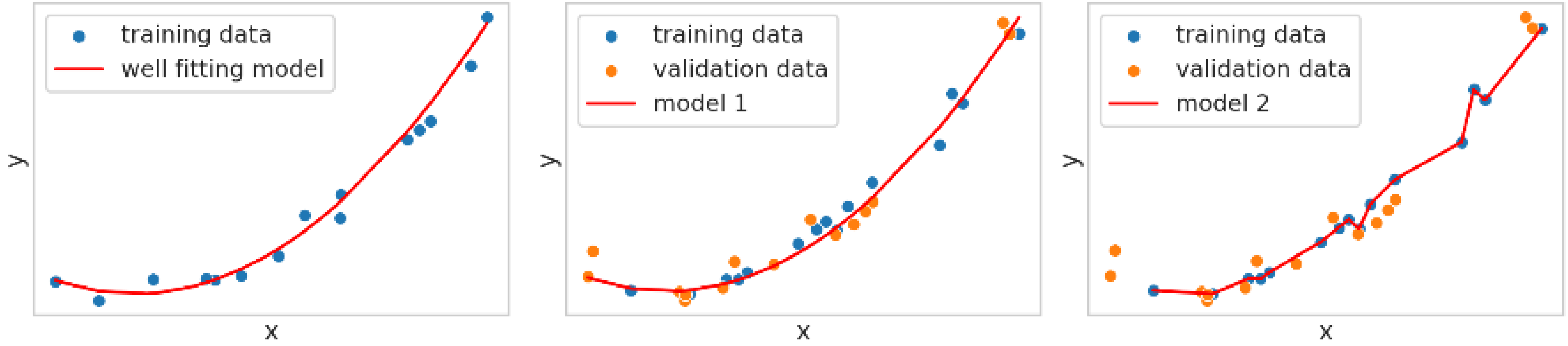




How to Avoid Overfitting

Identifying Overfitting

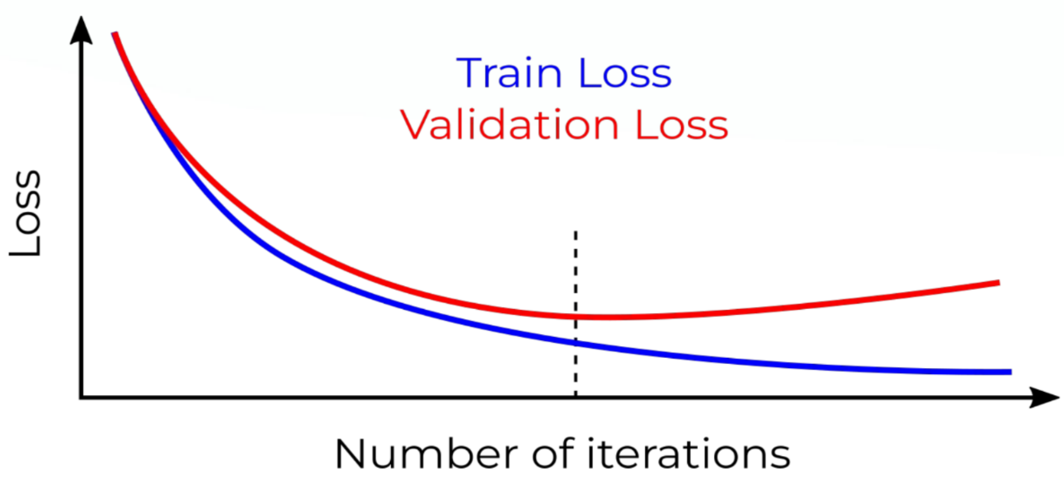
What do you think – which model performs better overall?



How to Avoid Overfitting

Training and validation data split

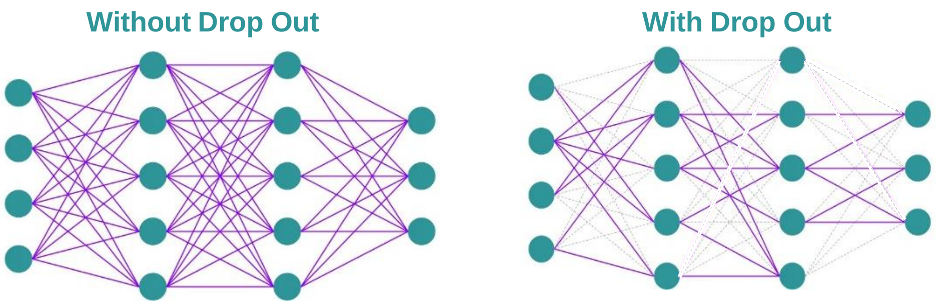
- Separate validation data (typically 10-20%)
- After training step, calculate the loss function using *only* the validation set



How to Avoid Overfitting

Regularization Technique

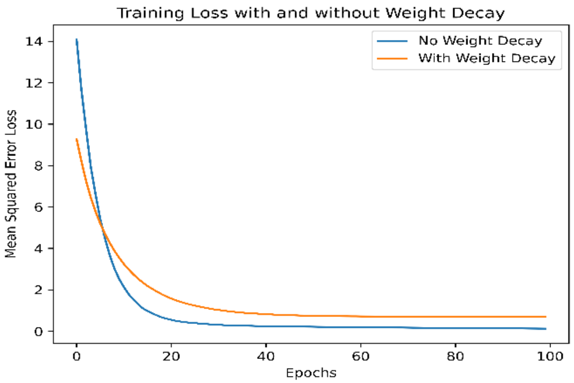
- Make use of dropout technique
  - Neurons are randomly dropped
  - It prevents over-dependent
  - Partial drop out could also be used
  - No dropout on output layer



- Make use of weight decay regularization

$$\text{RegularizedCost} = \text{Cost} + \lambda \sum_i w_i^2$$

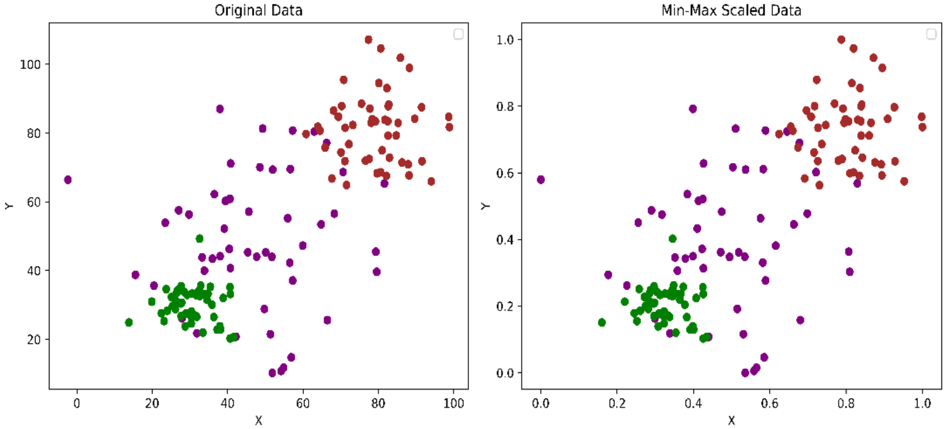
$$\text{updatedWeight}_i = \text{weight}_i - \alpha(grad_i + 2\lambda w_i)$$



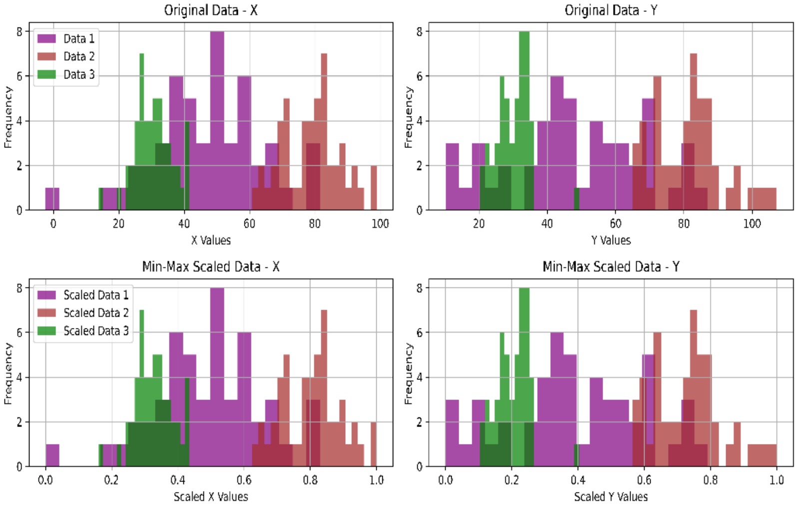
https://discuss.pytorch.org/t/how-does-sgd-weight-decay-work/33105/3

How to Improve Model

Normalization



- Ensure feature of similar scale
- Helps in convergence of model
- Indirectly prevents over-fitting



Common Normalization Methods

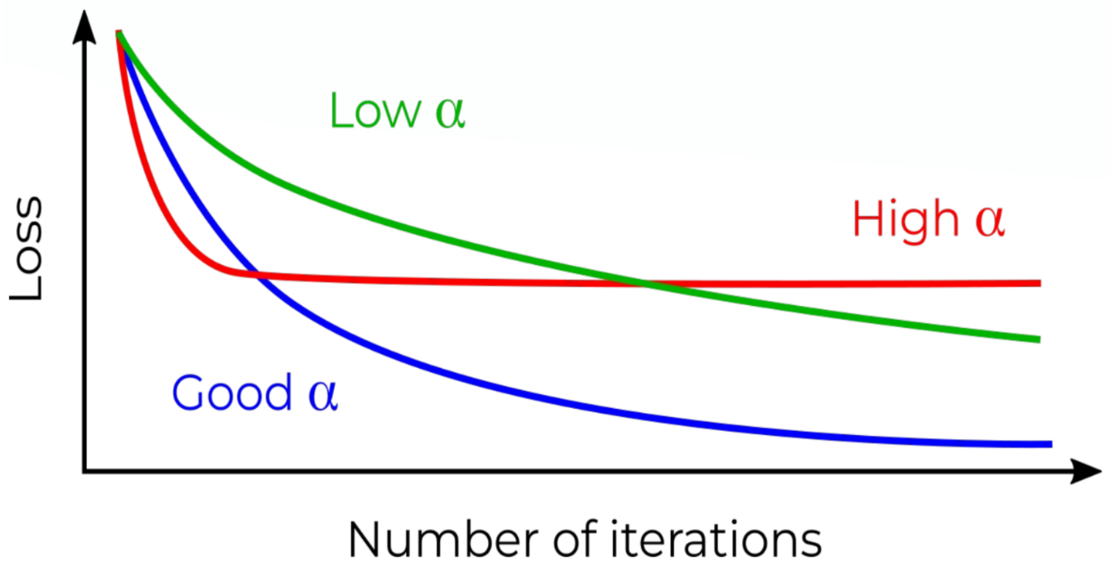
- Mix-Max Scaler
- Z-score Normalization
- Robust Scaler
- …

How to Improve Model

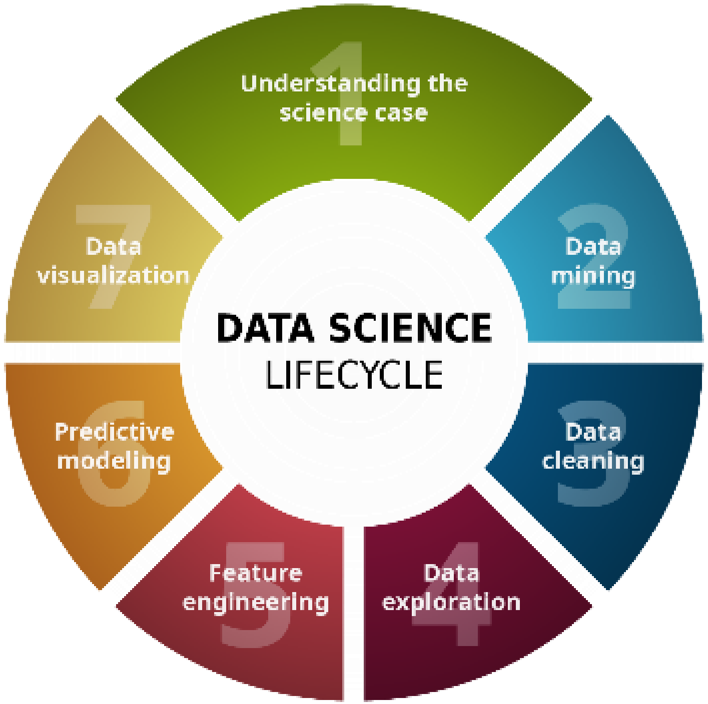
Hyper-parameter Optimization

Common Hyper-parameters

- Learning rate α
- Batch size
- Kernel size
- Epoch size
- Dropout rate
- …



Typical machine learning project cycle



- Common thinking: I will spend a lot of time in model development
- Reality: 90% of time is spent in data science parts
- Always set your code up for an iterative process
- Always follow best practices

Successful Machine Learning Projects

What do you need?

- Data that holds the necessary information and is of good quality
  - „Garbage in, garbage out“
  - Think in advance: How much data do you have? Can you obtain more?

- Model
  - Find the right model for your task (we will cover some in the course)
- Computational resources
  - Machine learning relies on GPUs
  - e.g. DKRZ Levante, JUWELS (HAICORE)

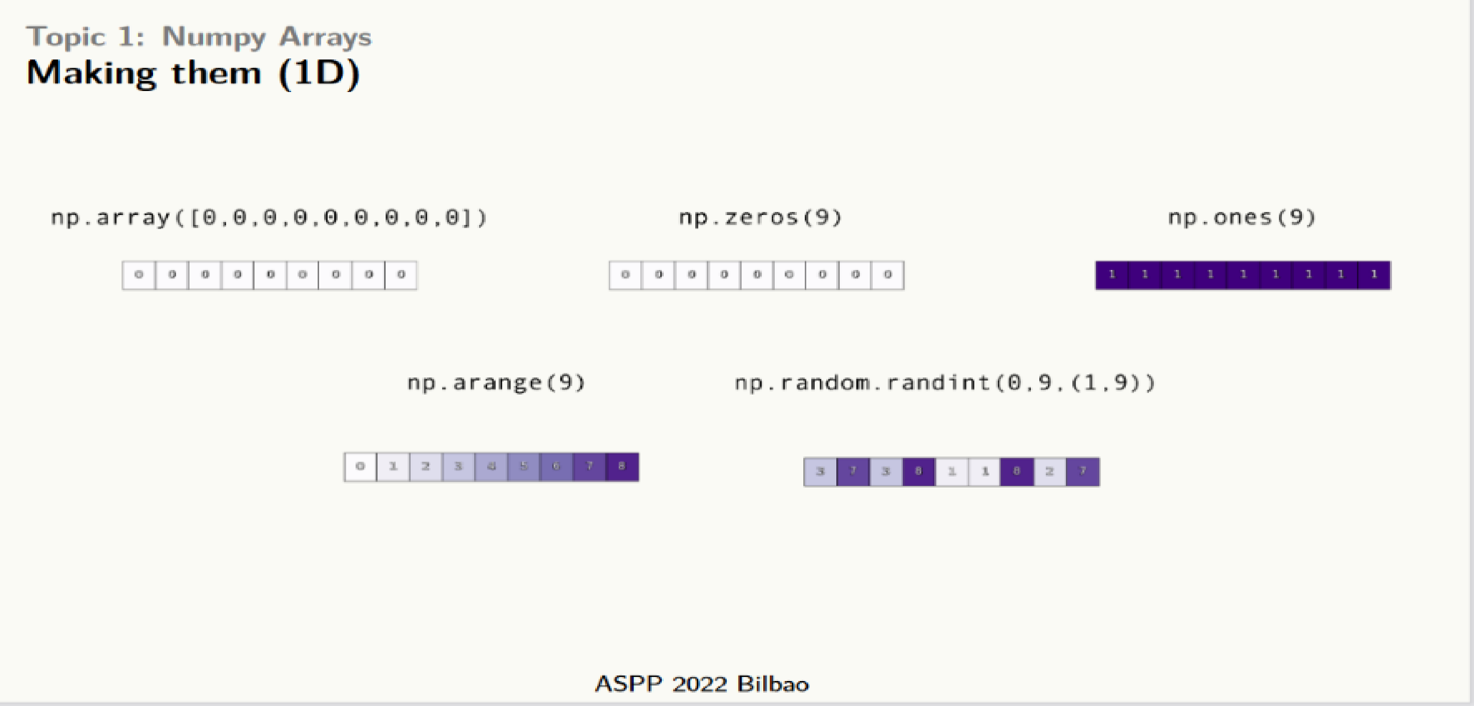
Questions?

Machine Learning and Python

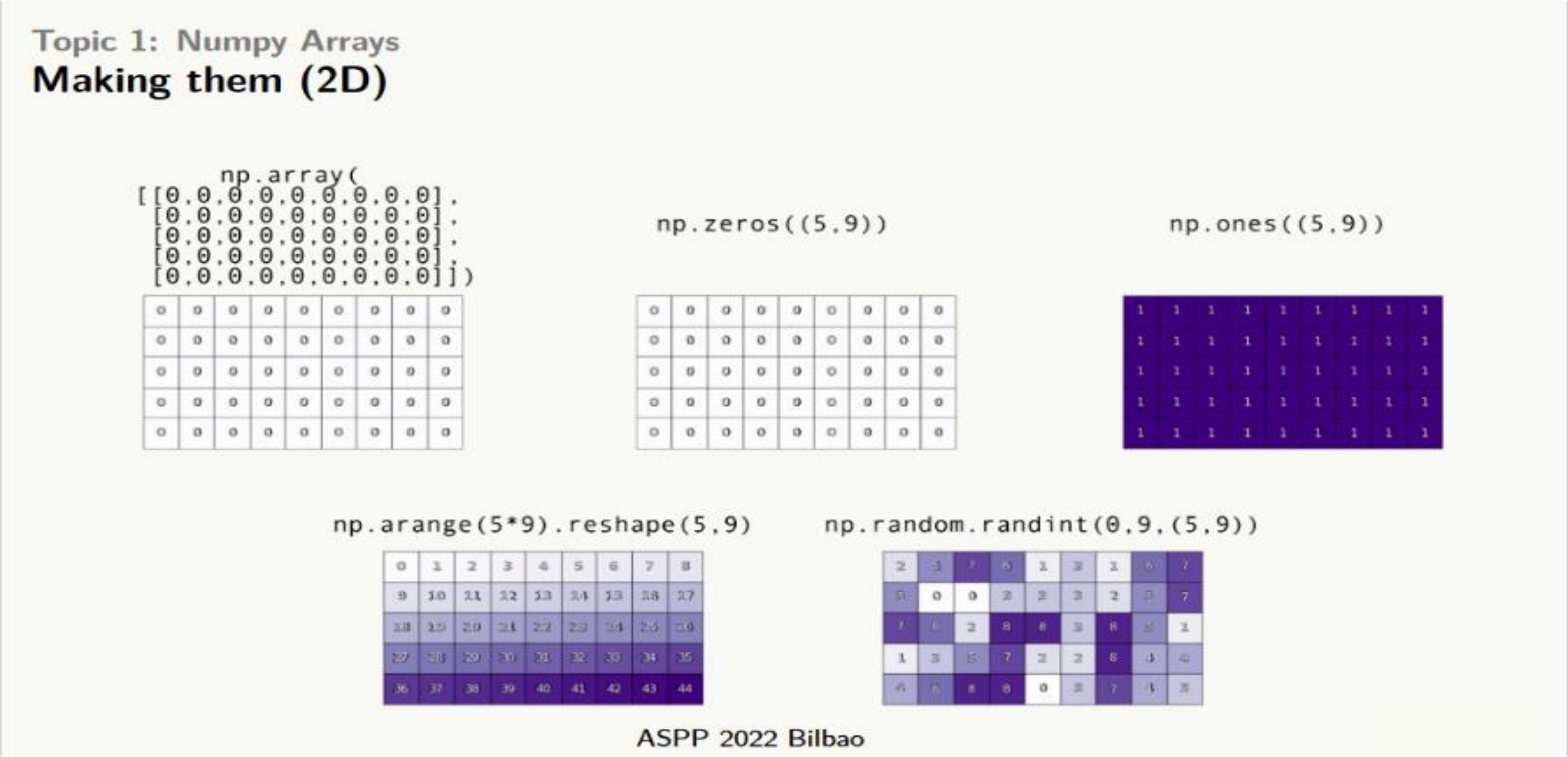
Libraries for data science and machine learning

https://devopedia.org/images/article/149/8470.1648284292.jpg

Numpy


Topic 1: Numpy Arrays
Making them (1D)
np.array([0,0,0,0,0,0,0,0,0])
0 0 0 0 0 0 0 0 0
np.zeros(9)
0 0 0 0 0 0 0 0 0
np.ones(9)
1 1 1 1 1 1 1 1 1
np.arange(9)
0 1 2 3 4 5 6 7 8
np.random.randint(0,9,(1,9))
3 7 3 8 1 1 8 2 7
ASPP 2022 Bilbao

Numpy



https://raw.githubusercontent.com/ASPP/2022-bilbao-advanced-numpy/master/ASPP_Numpy.pdf

xarray

A library for labelled datasets

- Extends numpy and panda: labelled multi-dimensional datasets
- Data model builds on netcdf standard → widely used for climate data
- Offers lazy loading – define all computations without loading the data from disk

https://docs.xarray.dev/en/stable/

xarray

Opening a netcdf file with xarray

```
ds = xr.open_dataset(nc_file)
```

[20]: xarray.Dataset

► Dimensions: (**time**: 12, bnds: 2, **depth**: 46, nodes_3d: 126859)

▼ Coordinates:

time	(time)	object	2293-01-31 23:59:59 ... 2293-12-...
depth	(depth)	float64	-0.0 10.0 20.0 ... 5.65e+03 5.9e+03

▼ Data variables:

time_bnds	(time, bnds)	object	...
thetao	(time, depth, nodes_3d)	float32	...

▼ Attributes:

CDI :	Climate Data Interface version 1.9.6 (http://mpimet.mpg.de/cdi)
Conventions :	CF-1.6
history :	Thu Jul 30 13:23:33 2020: cdo -s monmean -shifttime,-1sec /work/ba1066/a270124/esm-experiments/awicm_pism//RCP85/outdata/fesom//RCP85_fesom_thetao_22930101.nc /work/ba1066/a270124/esm-experiments/awicm_pism//RCP85/outdata/fesom//RCP85_fesom_thetao_22930101_monmean.nc
output_schedule :	unit: m first: 1 rate: 1

xarray

Create a DataArray

- List of precipitation values at different weather stations
- Annotate data array
  - Data
  - Coordinates
  - Dimensions
  - Name
  - Attributes
- → Much more descriptive than a standard numpy array

```
pr_data_xr = xr.DataArray(pr_data[:,0],
                          coords={"lon":("Station",pr_data[:,1]),
                                  "lat":("Station",pr_data[:,2])},
                          dims=["Station"],
                          name="Precipitation",
                          attrs={"units":"mm",
                                 "coords":"lon lat"})
```

Questions?