ETH zürich

SPACE GEODESY

Machine learning strategies in geodetic applications

Benedikt Soja

August 29, 2025, NKG Summer School

My background

TECHNISCHE UNIVERSITÄT WIEN

GFZ Helmholtz Centre Potsdam

NASA JPL

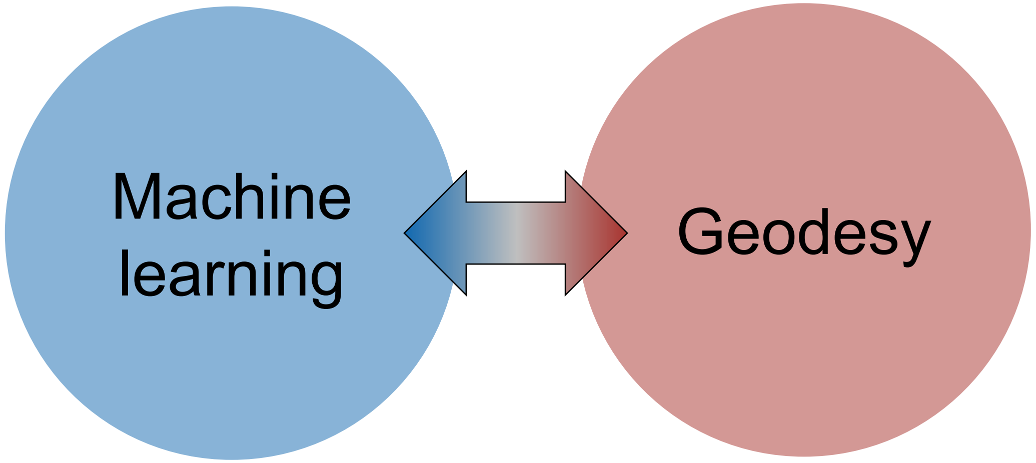
ETH zürich

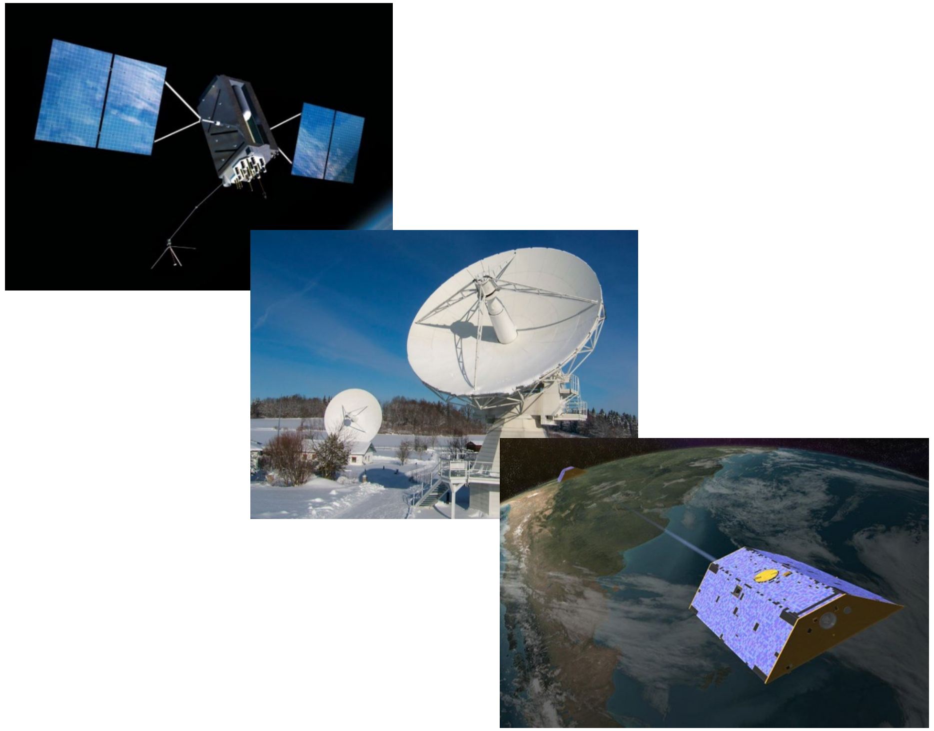
SPACE GEODESY

MSc	2013	PhD	2016	PostDoc	2020	Asst. Prof. (Tenure Track)

Chair of Space Geodesy @ ETH Zurich

- Established in April 2020
- Research focus: machine learning in geodesy
- GNSS, VLBI, satellite gravimetry

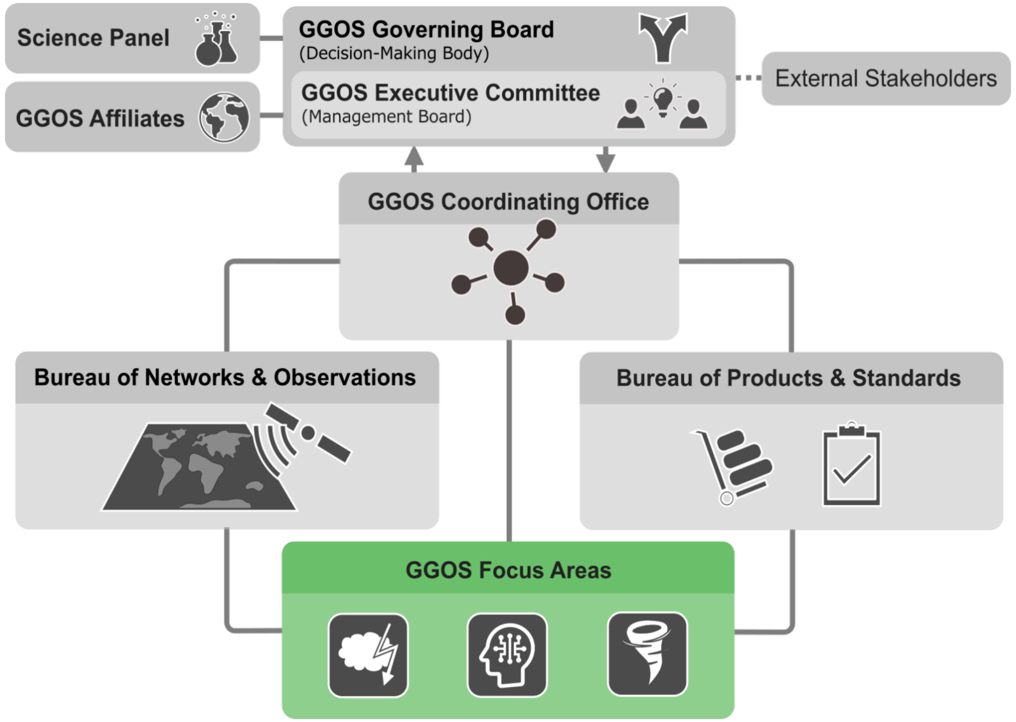




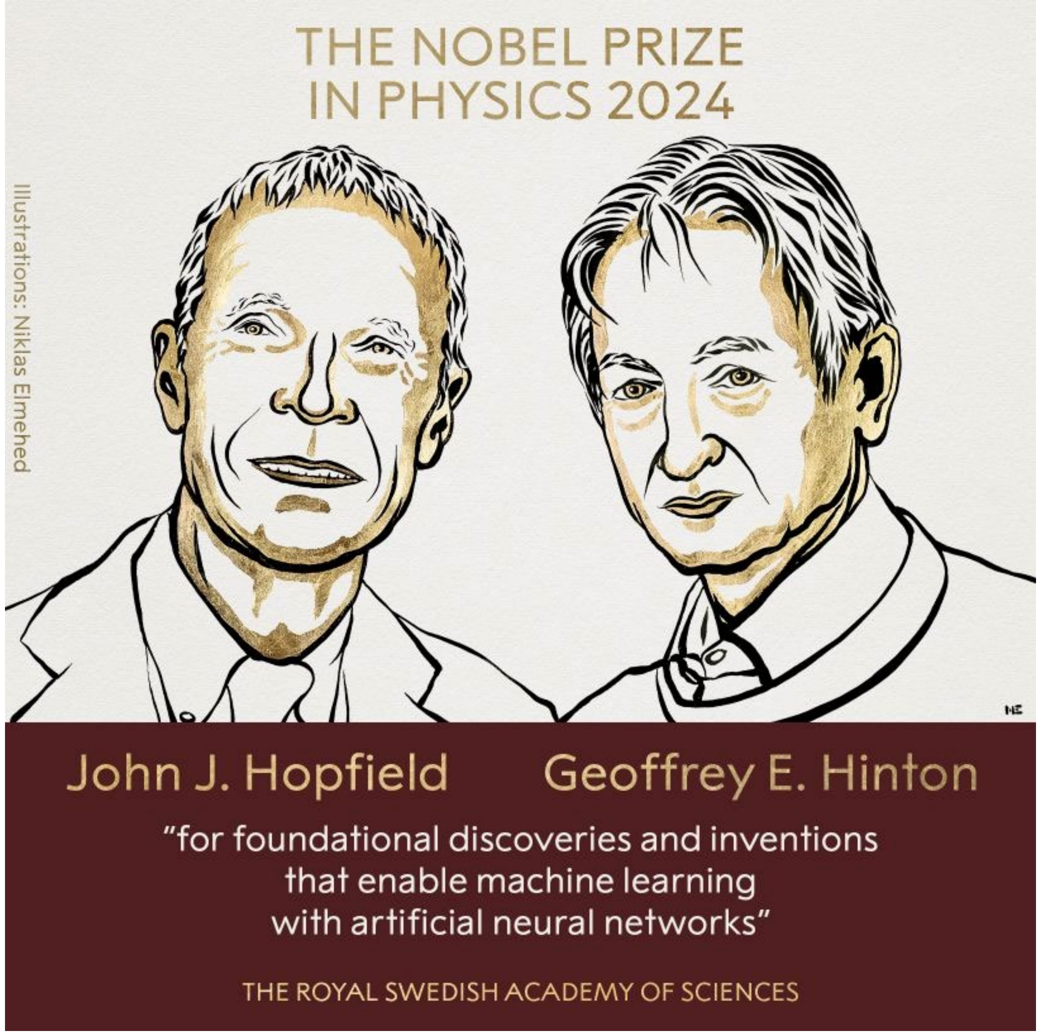
GGOS Focus Area: AI for Geodesy

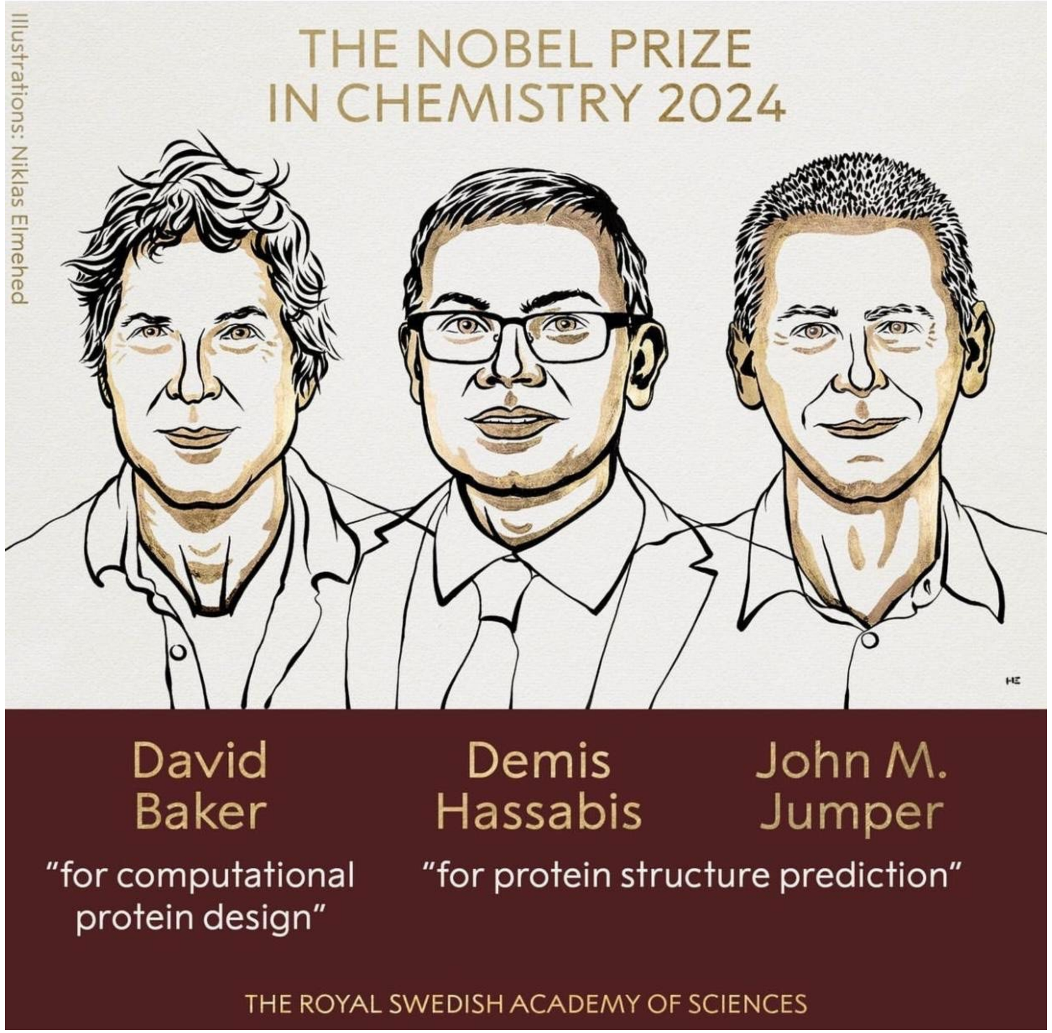
- Chair: Prof. Dr. Benedikt Soja (ETH Zurich, Switzerland)
- Vice-chair: Dr. Maria Kaselimi (NTUA, Greece)

- **Objective**: develop & evaluate improved geodetic products based on AI and machine learning
  - Higher accuracy and resolutions (spatial/temporal)
  - Improved real-time and prediction quality
  - Trustworthy & interpretable



- Four **joint study groups**
  - 100+ Members
  - 50+ institutions
  - 10+ countries


THE NOBEL PRIZE
IN PHYSICS 2024
Illustrations: Niklas Elmehed
John J. Hopfield
Geoffrey E. Hinton
"for foundational discoveries and inventions
that enable machine learning
with artificial neural networks"
THE ROYAL SWEDISH ACADEMY OF SCIENCES

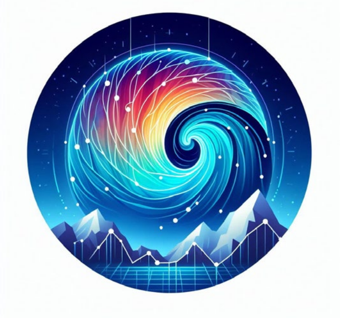

THE NOBEL PRIZE
IN CHEMISTRY 2024
Illustrations: Niklas Elmehed
David
Baker
Demis
Hassabis
John M.
Jumper
"for computational
protein design"
"for protein structure prediction"
THE ROYAL SWEDISH ACADEMY OF SCIENCES

AI for Science has emerged as an overarching trend

- Big tech companies (Google, NVIDIA, etc.) invest heavily in AI for Science, including:
  - AI for medicine/health
  - AI for chemistry/material science
  - AI for Earth system modeling (climate, weather, etc.)

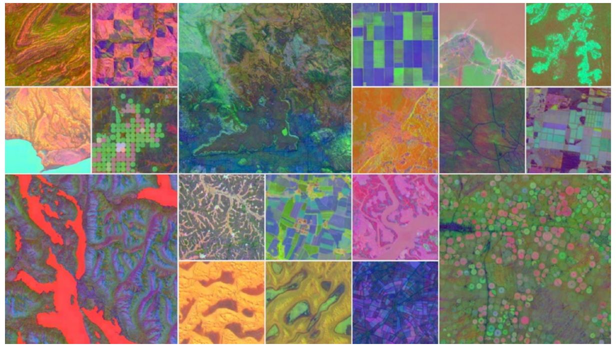
- The “magic ingredient” is typically the **amount and quality of data**
  - Model choices are already very mature
  - Computation not a limitation for such companies

Aurora: A Foundation Model of the Atmosphere



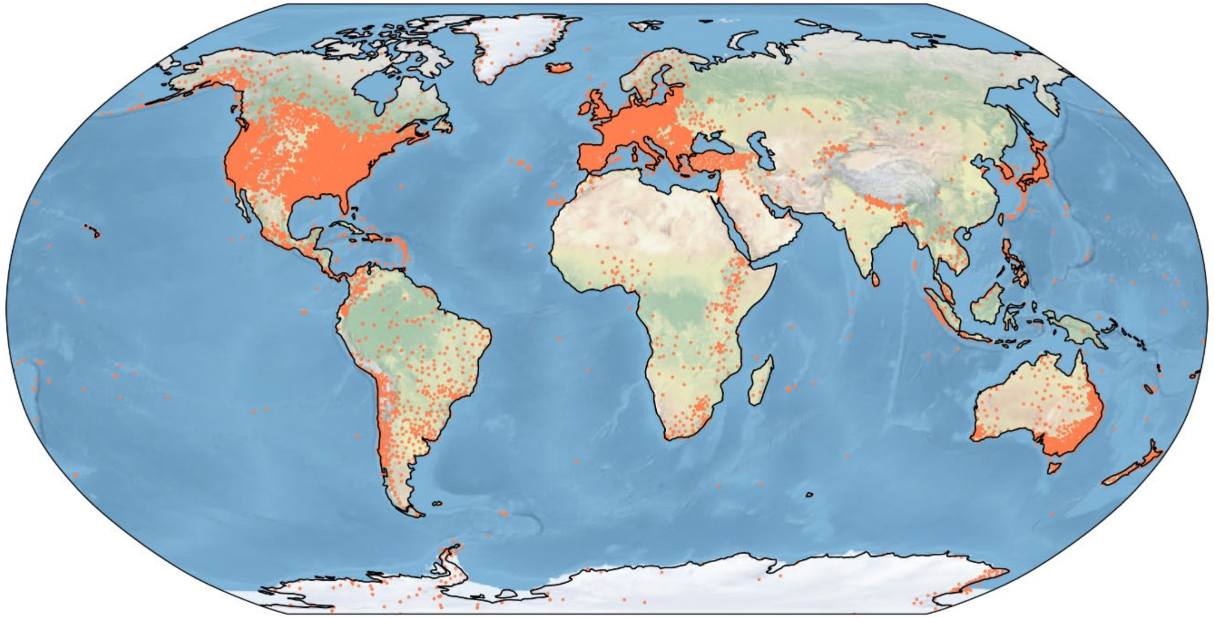
Microsoft

AlphaEarth Foundations



Focus on data is a great opportunity for geodesy!

- Huge increase in data volume from GNSS stations, InSAR, altimetry, etc.
- Geodetic data of very high quality (“mm-level”)
- Auxiliary data: weather, climate, environmental models, etc.



20’000 GNSS stations

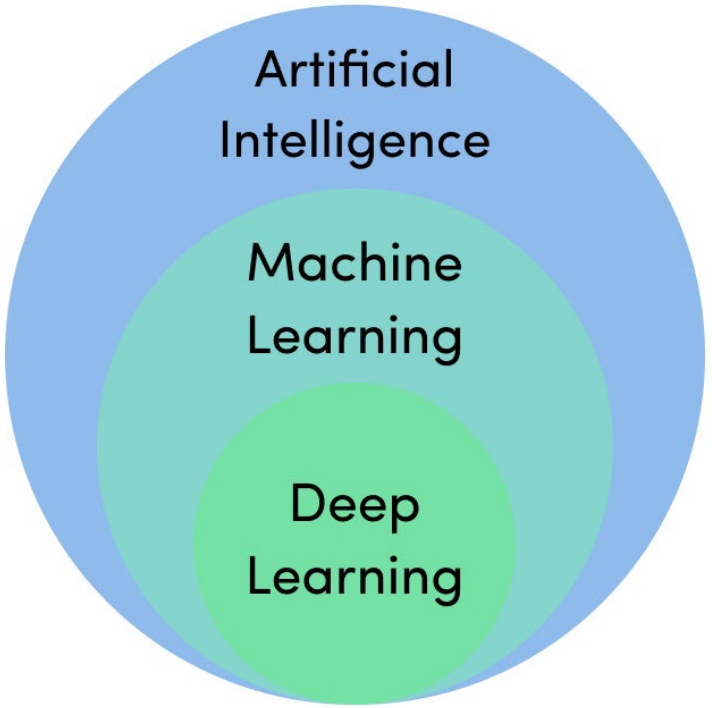
Outline

- Introduction to machine learning
- Types of machine learning problems
- Selected machine learning algorithms
- Machine learning strategies in geodetic applications

Introduction to machine learning

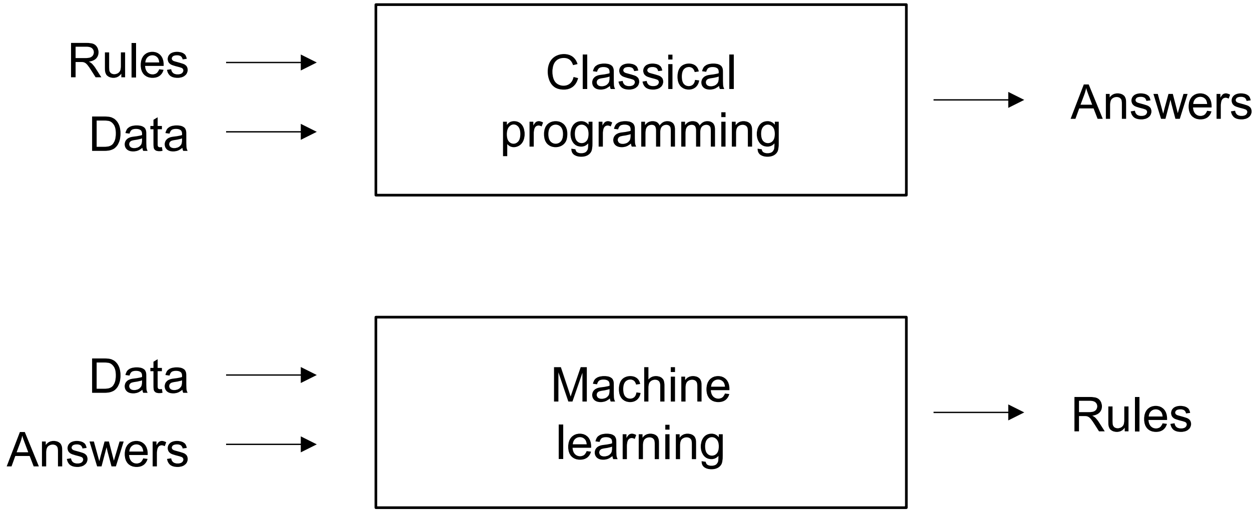
AI vs. machine learning

- Artificial intelligence (AI): any technique that enables machines to mimic human intelligence
- Examples:
  - machine learning
  - evolutionary strategies
  - rule-based chatbots
  - …



What is machine learning?

- Arthur Samuel (1959): giving computers the ability to learn without being explicitly programmed



Example: weather prediction

- Classical approach to numerical weather prediction

Weather today → **Classical programming** → Weather tomorrow

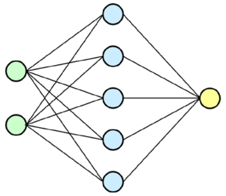
Physical knowledge → **Classical programming**

Example: weather prediction

- Training a machine learning model using decades of weather data

Weather today → Machine learning → Weather prediction model

Weather tomorrow →

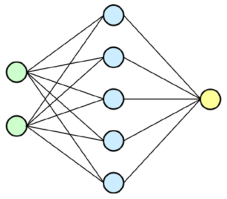


Example: weather prediction

- Training a machine learning model using decades of weather data

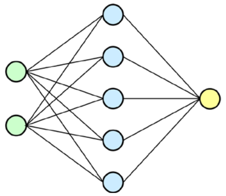
Weather today → Machine learning → Weather prediction model

Weather tomorrow → Machine learning

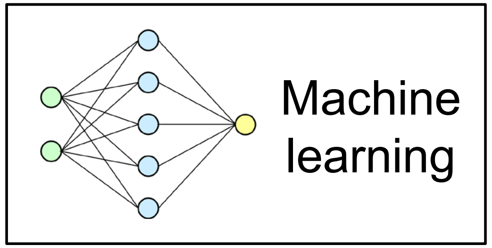


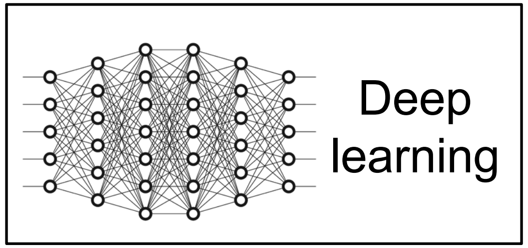
- Applying the machine learning model operationally

Weather today → Weather prediction model → Weather tomorrow

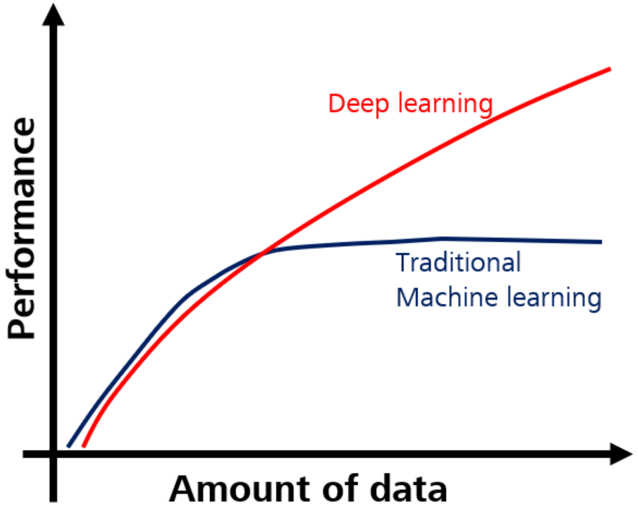


What is deep learning?





- Sub-category of machine learning making the computation of multi-layer neural networks feasible
- No clear separation, but some generally accepted characteristics
- Algorithms
  - ML: shallow neural networks, decision trees, support vector machines, etc.
  - DL: deep neural networks (3+ hidden layers)
- Data
  - ML: manual feature extraction
  - DL: “big” data, automatic feature extraction
- Number of parameters
  - ML: neural networks with 10s-100s of parameters
  - DL: extreme case GPT-4 with 1,000,000,000,000 (1 trillion) parameters



apeer.com

Types of machine learning problems

Main machine learning categories

- Unsupervised learning
  - Methods that automatically detect patterns in data without using labels
  - Examples: principal component analysis, clustering, autoencoders, …
- Supervised learning
  - Methods that automatically learn an input-output relationship based on example input-output pairs
  - Examples: support vector machines, decision trees, random forests, neural networks, …
- And many more variants, e.g., reinforcement learning, self-supervision

Unsupervised learning

- Methods that automatically detect patterns in data without using labels
- Examples: principal component analysis, clustering, autoencoders, …

Principal Components Analysis

- Eigendecomposition of the covariance matrix $$Cv = \lambda v$$
- Principle components maximizing the variance

- Dimensionality reduction
  - Useful as a pro-processing step
  - Often used before applying other machine learning algorithms

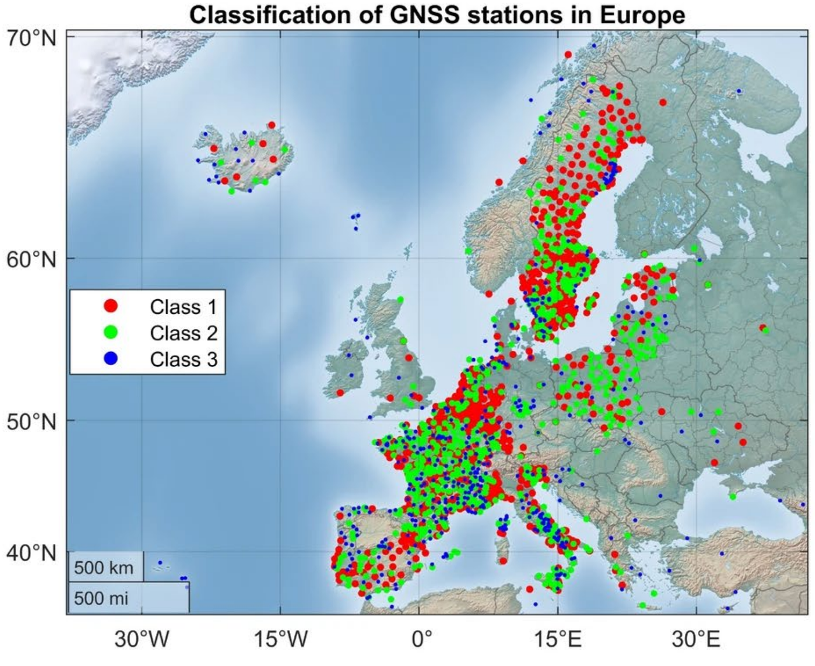
- Is it machine learning?
  - Up to debate
- Unsupervised?
  - Yes, no human input required (except when removing PCs)

Clustering

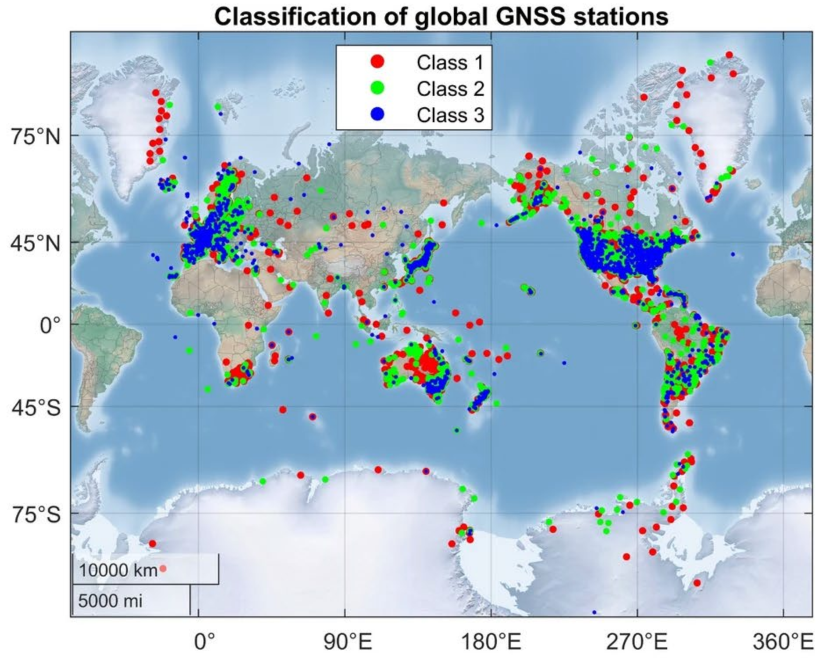
- Automatically creating groups of data points with similar properties
- Most famous algorithm: k-means clustering
  - Iterative optimization problem
  - Method:
    - find k clusters centers
    - assign data points to nearest cluster
    - minimize squared distances to the cluster center
- Other algorithms are based on probability distributions or density

Clustering example

- Clustering (Hierarchy, K-means, or Fuzzy C-means) to divide GNSS stations into 3 classes
- Clusters form the basis for a following classification task



(a)

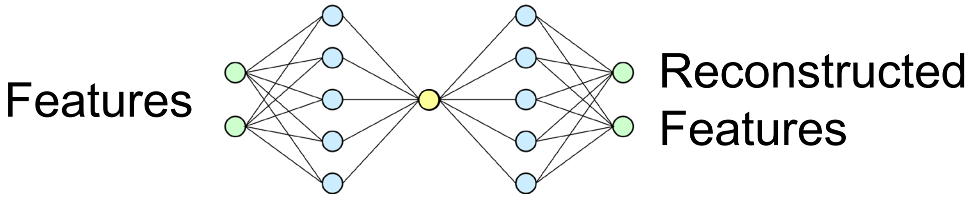


(b)

Le, N., Männel, B., Bui, L.K. *et al.* Classifying continuous GNSS stations using integrated machine learning. *GPS Solut* **29**, 44 (2025). https://doi.org/10.1007/s10291-024-01797-2

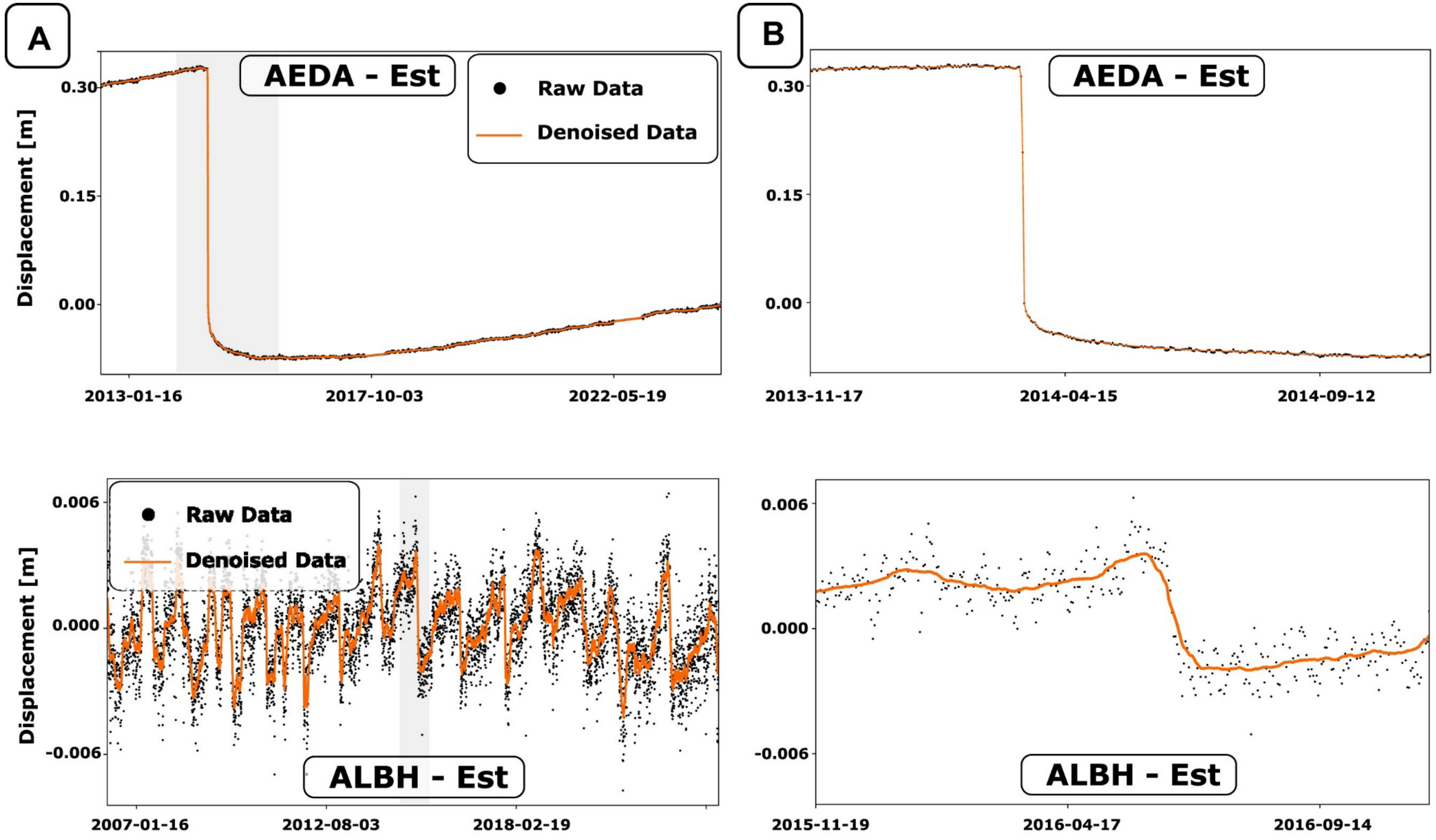
Autoencoders

- Goal: compress data into a lower-dimensional latent space and reconstruct it
- Structure:
  - Encoder: maps input features → latent representation
  - Decoder: maps latent representation → reconstructed features
- Training objective: minimize reconstruction error (e.g., MSE)
- Applications:
  - Dimensionality reduction (non-linear PCA)
  - Denoising data
  - Anomaly detection
  - Generating new samples



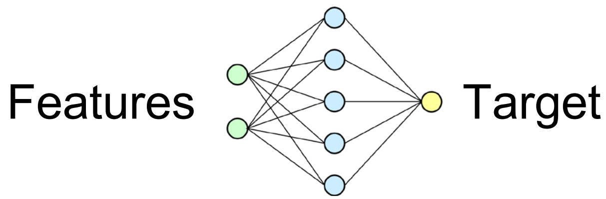
Autoencoder example

- Learn both GNSS station position trajectory and realistic noise behavior for tectonic applications
  → denoising of GNSS time series



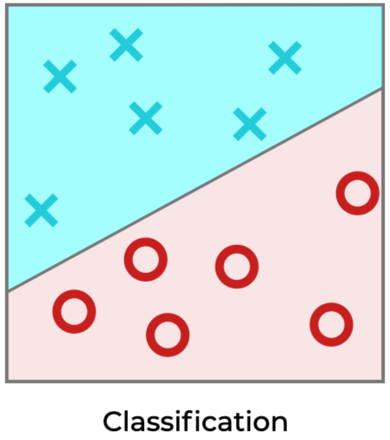
G. Mastella, J. Bedford, F. Corbi, F. Funiciello, *et al.*, Denoising daily displacement GNSS time series using deep neural networks in a near real-time framing: a single-station method, *Geophysical Journal International*, https://doi.org/10.1093/gji/ggaf207

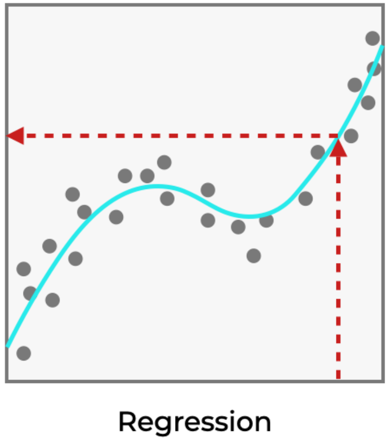
Supervised machine learning



- Input data: features
- Output data: labels/target
- Measure of model performance: “loss”
- Estimation of model parameters with backpropagation

- Classification problems
  – Predict a discrete label (class, category, …)
- Regression problems
  – Predict a continuous label





sharpsightlabs.com

Supervised machine learning

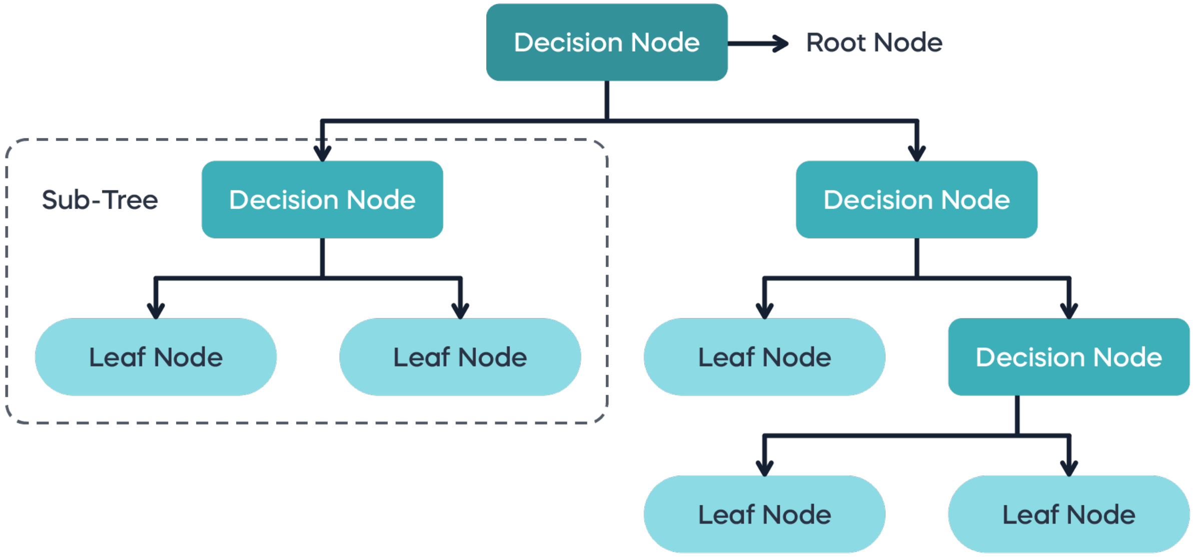
- Training:
  - Initialize model
  - Compute label predictions by propagating features through model
  - Compare label predictions with actual labels → compute loss
  - Update model to reduce loss
  - Repeat

- Prediction:
  - Compute label prediction based on ***unseen*** data

Selected machine learning algorithms

Decision tree principle

- Predict a value based on several input variables
- Starting with the root, then splitting off into several branches, finally reaching the leaves
- Each node splits the data according to a certain criteria based on a single feature
- Classification trees: leaves represent discrete classes
- Regression trees: high number of leaves → close to continuous



365datascience.com

Properties of decision trees

Pros:

- No scaling needed
- Interpretable (unlike most other machine learning algorithms)

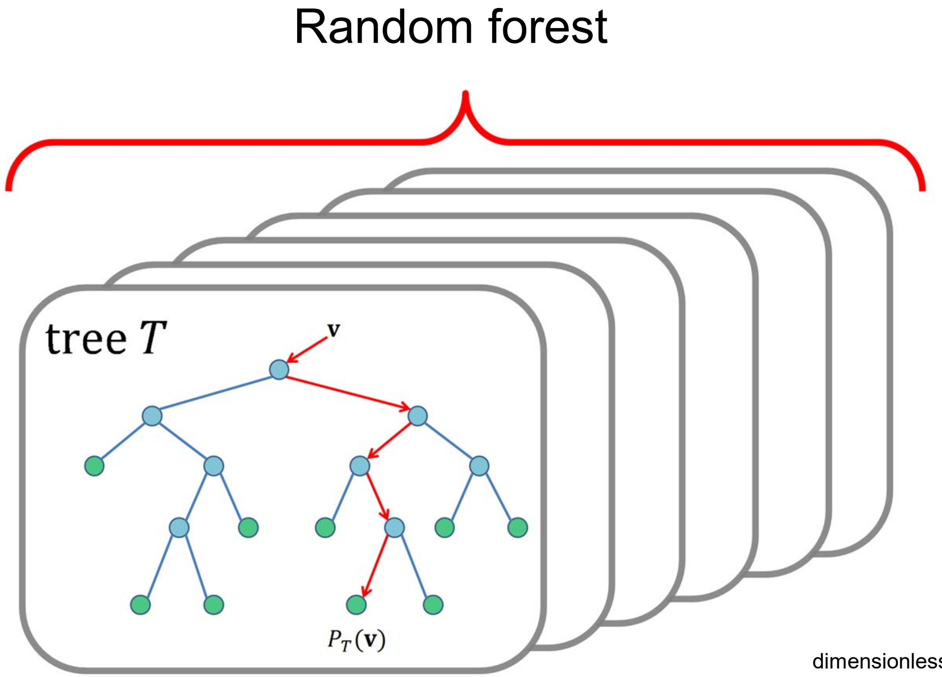
Cons:

- Perform worse than other prediction algorithms
- Very sensitive to input data distribution

→ random forests address some of the disadvantages

Random forests

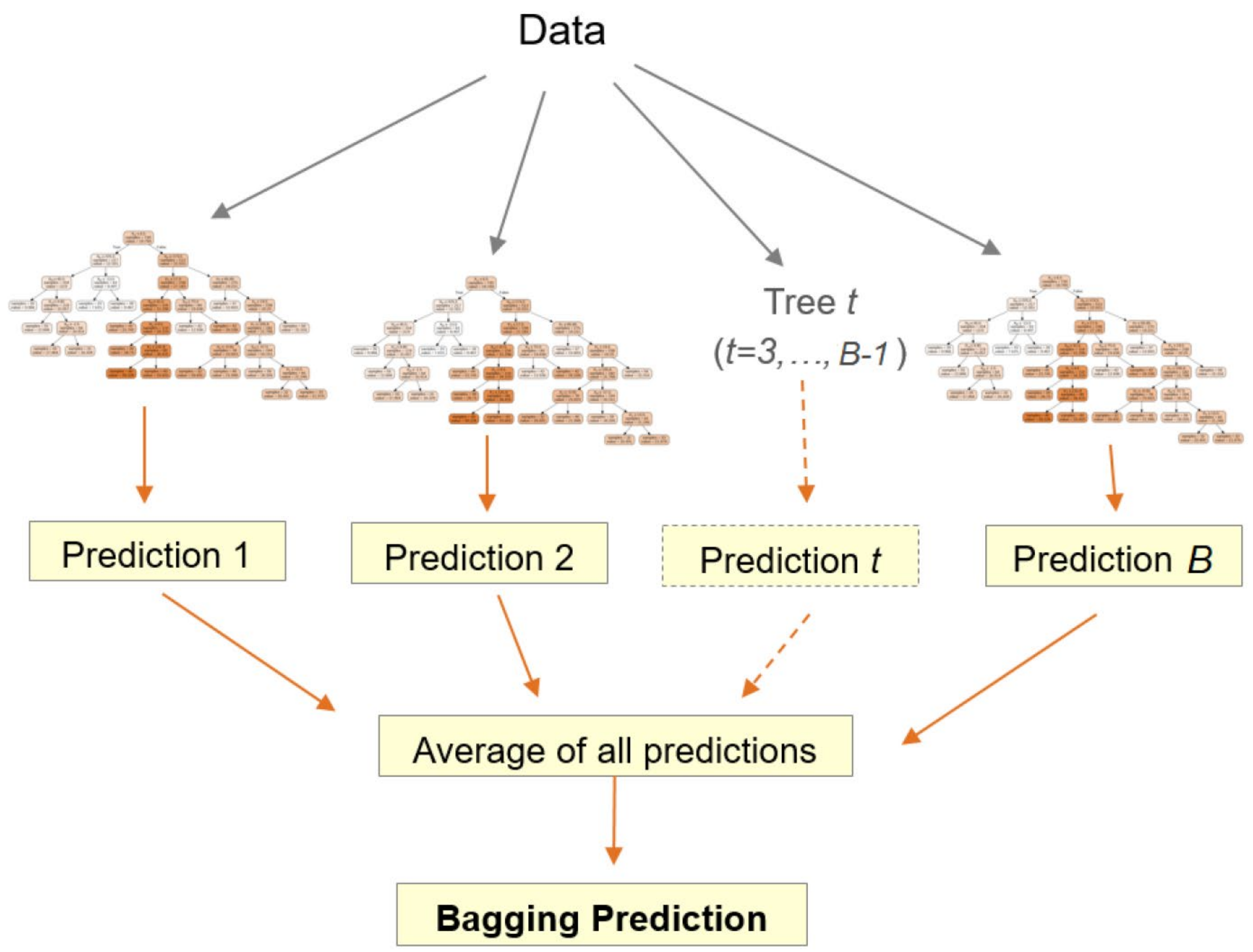
- Random forests consist of an ensemble of decision trees
- Individual trees are created randomly (bootstrapping)
- Final result: results from all trees are aggregated (bagging)
- Advantage: errors of individual trees are averaged out



dimensionless.in

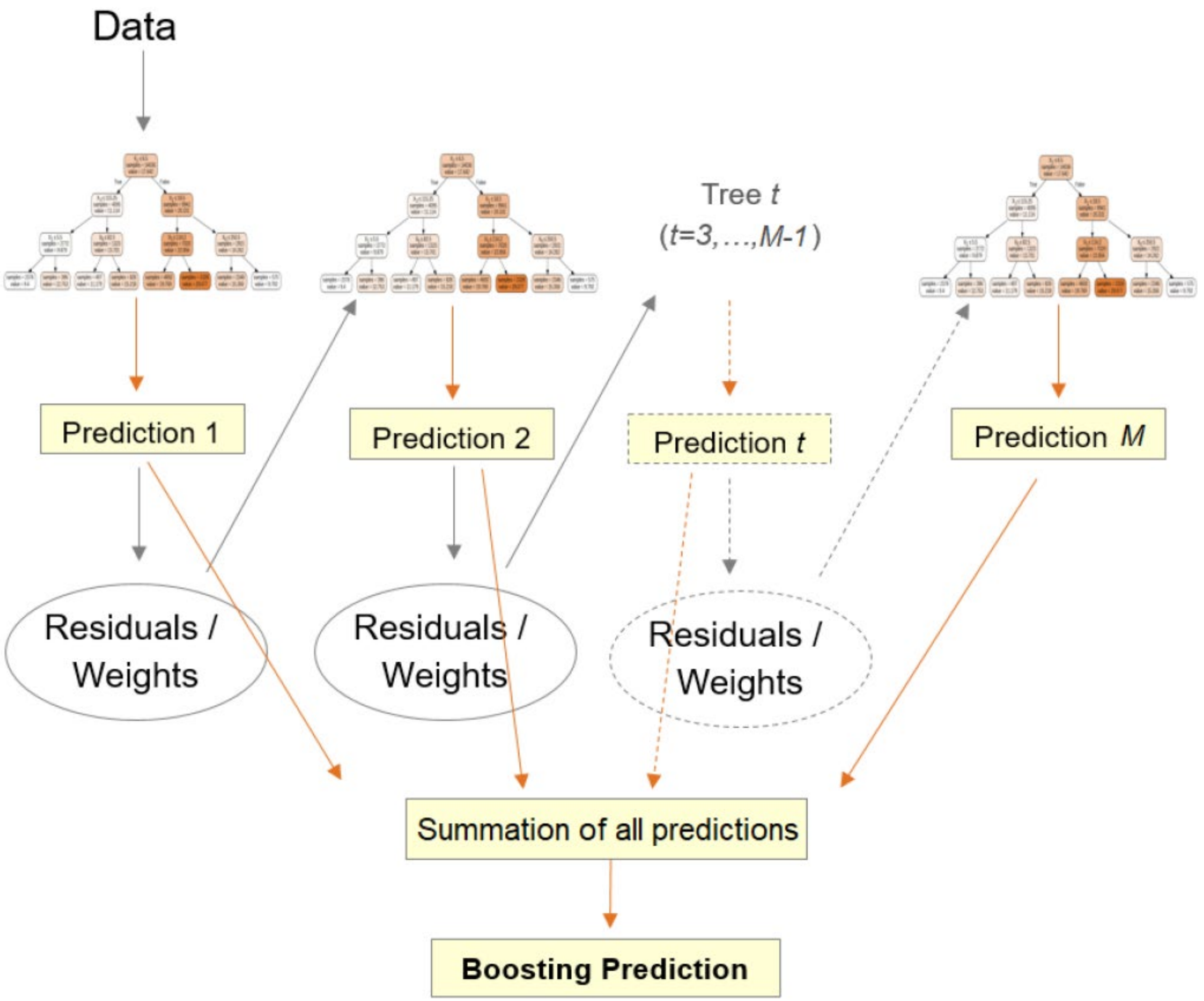
Bagging principle of random forests

- Trees are grown individually based on random data samples and features

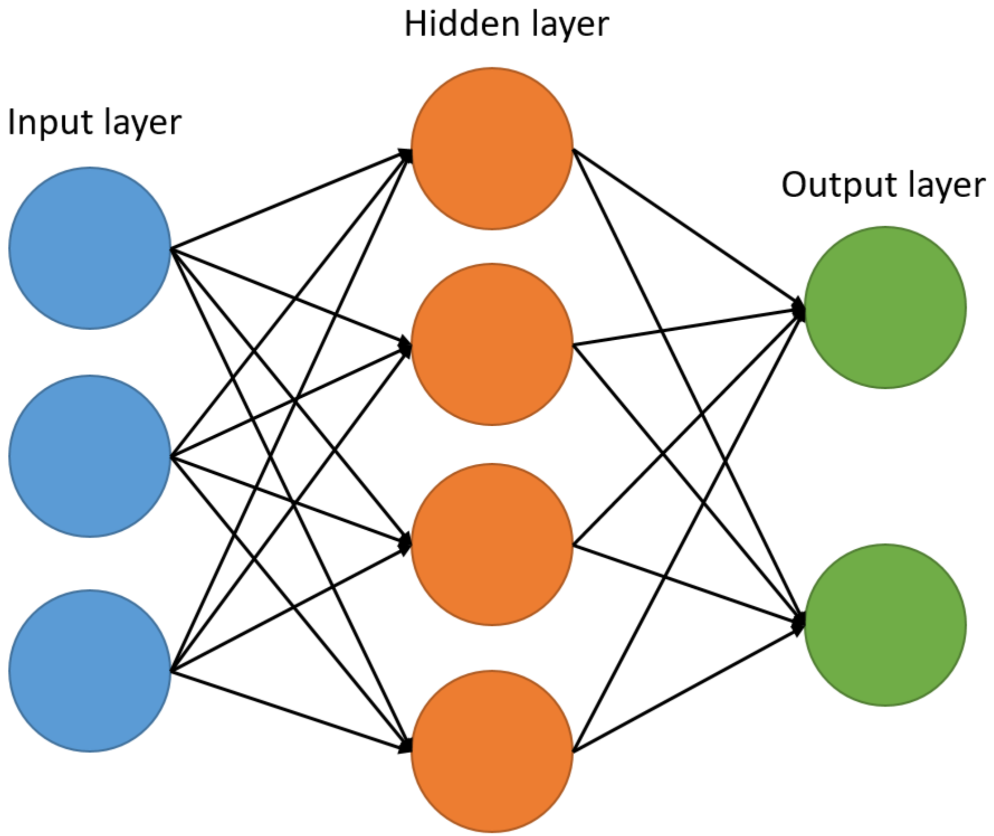


Boosting trees

- Trees are grown sequentially
- Adaptive boosting (AdaBoost)
  - Increasingly higher weights are applied to samples more difficult to predict
- Gradient boosting (e.g., XGBoost)
  - Additional trees are based on residuals from previous trees



Feedforward NNs



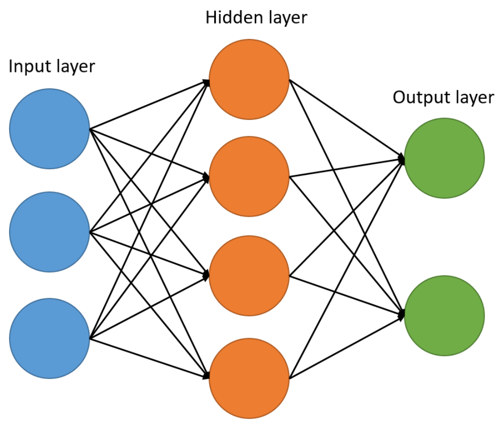
- Simplest form of NNs
- Information flows from input to output – no cycles
- Minimum: input & output layers
- Optional: hidden layer(s)
- Each edge has a weight
  - Unknown parameters of the NN
- At each node:
  - Sum of the weighted inputs calculated
  - Activation functions applied
- Bias nodes

- Examples: single-layer or multi-layer perceptrons, convolutional neural networks

Activation functions

- Needs to be differentiable!
- Examples
  - Linear function $f(x) = x$
  - Sigmoid function (logistical function) $f(x) = \frac{1}{1 + e^{-x}}$
  - Tangens hyperbolicus $f(x) = \tanh x$
  - Rectified linear unit (ReLU) $f(x) = \max(0, x)$
  - … and many more

Backpropagation



- Algorithm to determine weights of a neural network

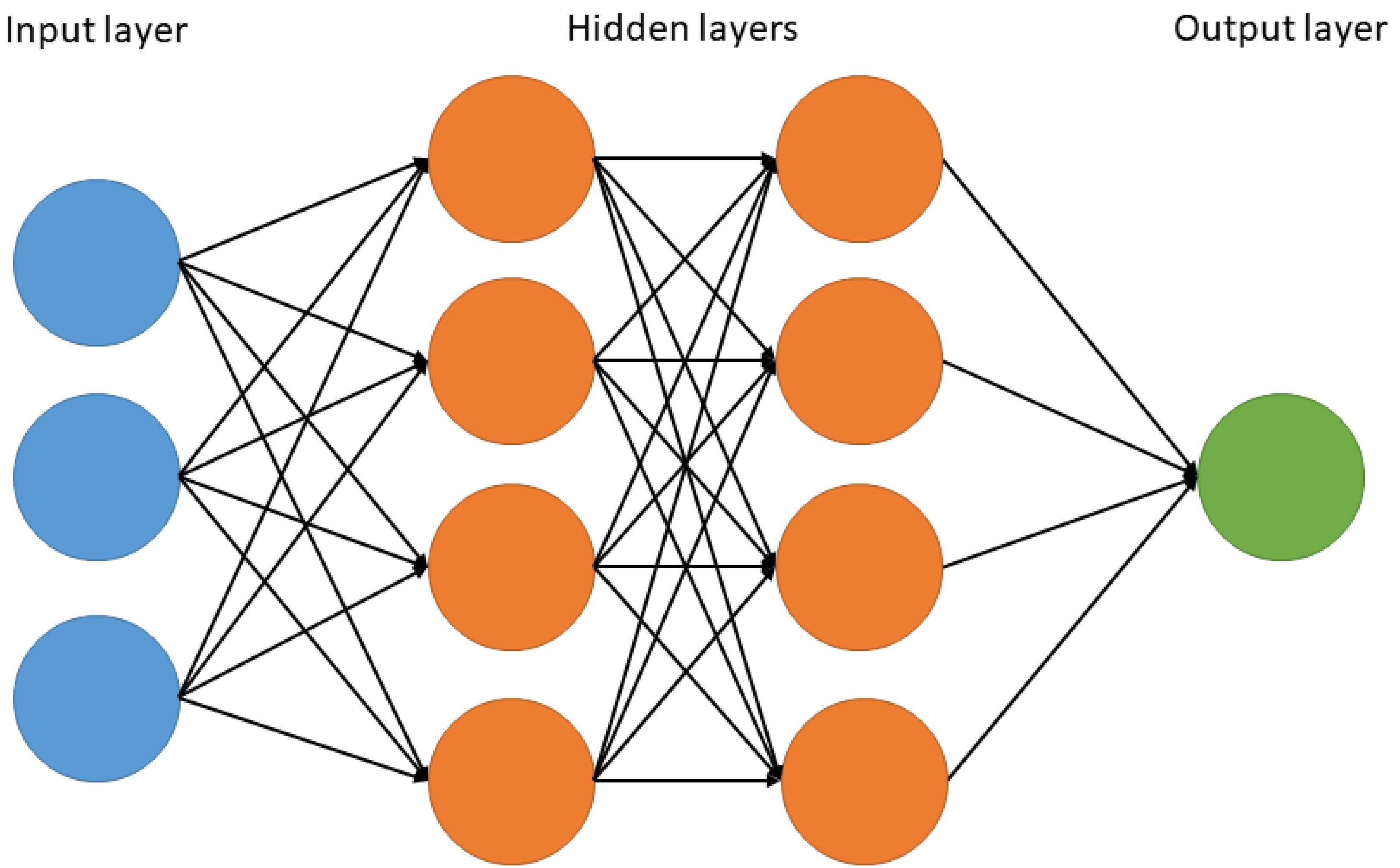
1. Inputs are propagated forward through the net
2. Predicted outputs are compared with actual outputs → computation of loss
3. Error is propagated backward through the net
   – Weights are adjusted based on their impact on the error
   – Impact is determined by partial derivatives / gradients of loss functions w.r.t. weights

Training a neural net with backpropagation

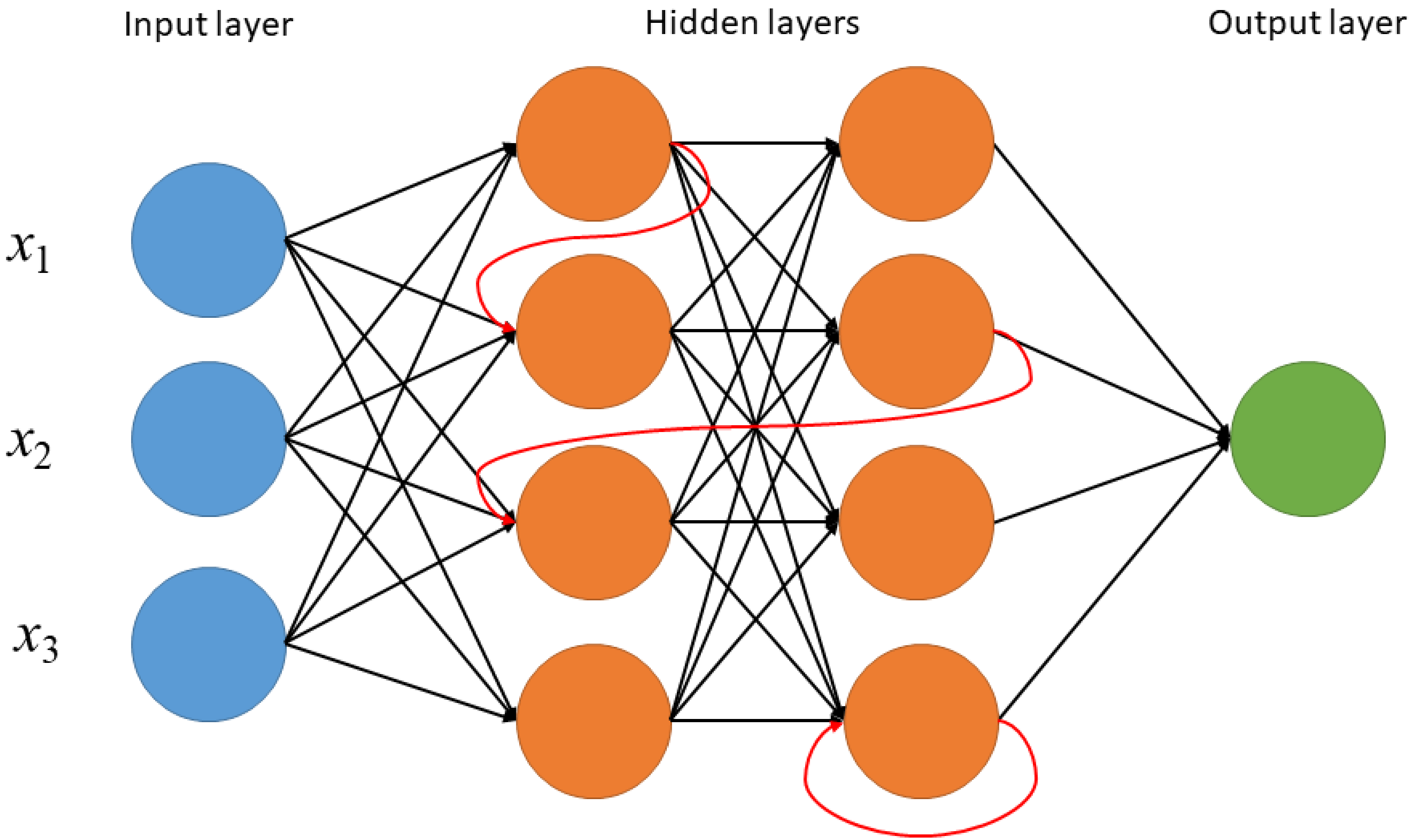
- Determination of weights based on gradients while minimizing loss: optimization problem
- Commonly used methods:
  - Gradient descent
  - Levenberg-Marquardt
- Feedforward NN: efficient backpropagation with simple matrix multiplications

- General problem: global vs. local minima
  - Potential solution: more sophisticated initialization of weights
- Problem with deep neural networks (high number of layers): vanishing/exploding gradients
  - Potential solution for vanishing gradient: specific neural networks (e.g., LSTM)
  - Potential solution for exploding gradient: activation functions that cannot “explode”, gradient clipping

Feedforward NNs

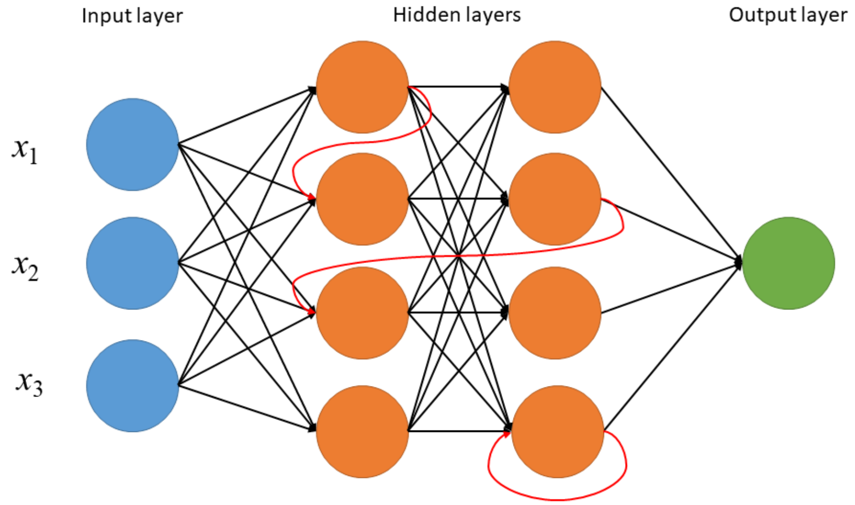


Recurrent neural networks



Recurrent neural networks

- Input is typically a time series → input nodes correspond to temporal sequence
- RNNs can detect temporal patterns and compute predictions
- RNNs have loops and other additional connections compared to feedforward NNs
  - Direct, indirect, lateral feedback
- Training: backpropagation through time (BPTT; generalized backprop algorithm) & gradient descent

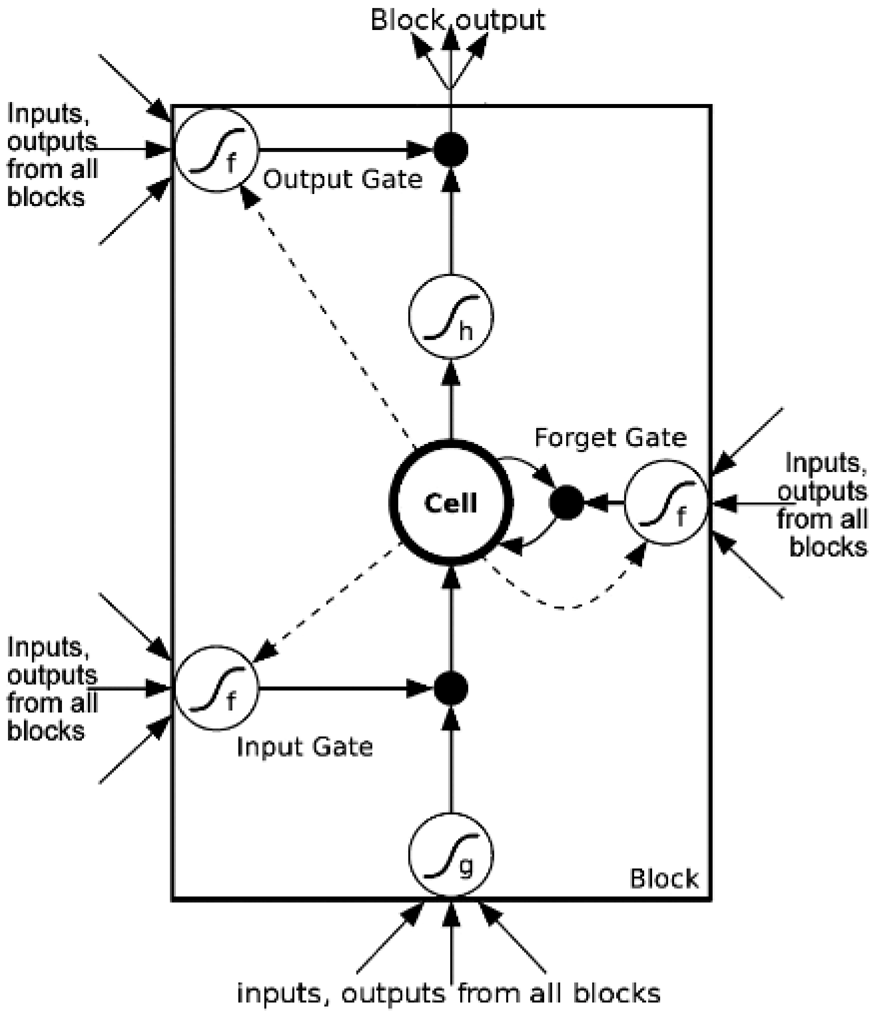


Long short-term memory (LSTM)

- Problem of RNNs: vanishing or exploding gradients when training multi-layer RNNs
- LSTMs solve the vanishing gradient problem

- Activations corresponds to short-term memory
- Weights correspond to long-term memory

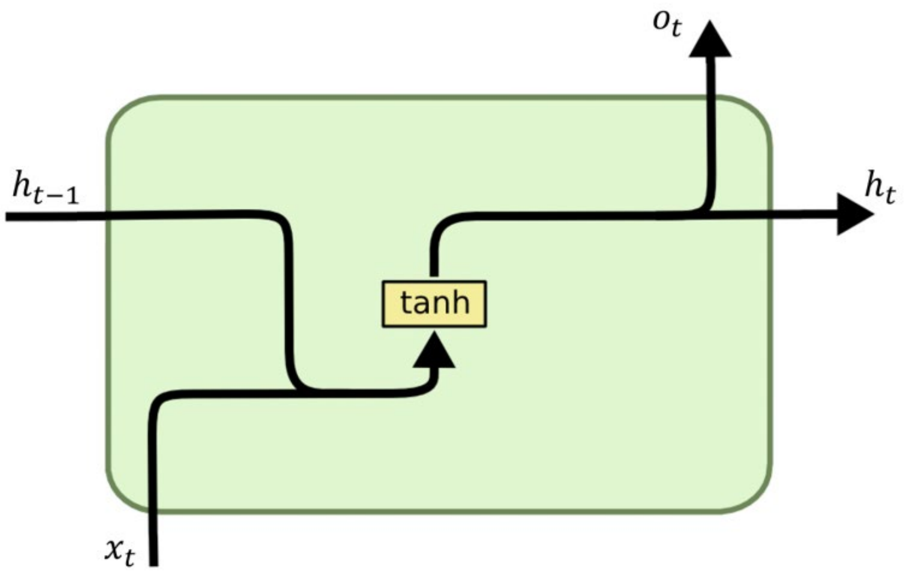
- LSTMs typically based on cells with 3 gates:
  - Input gate
  - Output gate
  - Forget gate

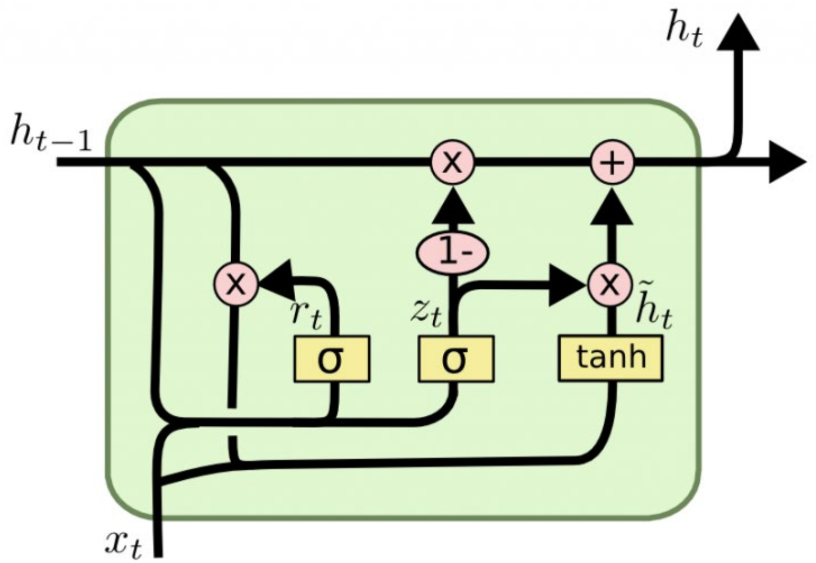


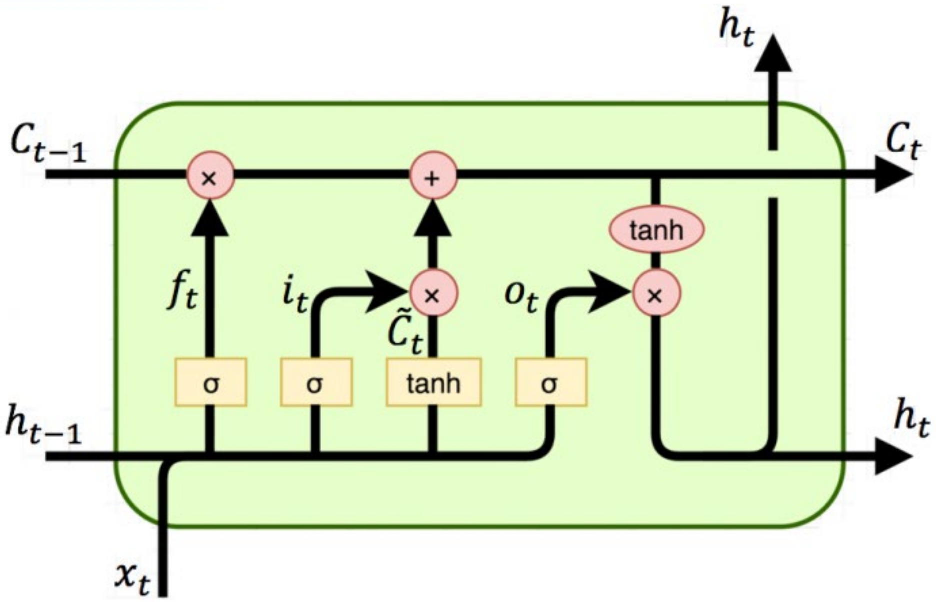
Gated recurrent units (GRU)

- Similar to LSTM, but with fewer parameters
- Just 2 gates:
  - Reset
  - Update
- GRU also address the vanishing gradient problem
- LSTMs are more powerful, but GRUs are easier to train
  - Often GRUs provide similar performance, but are much more efficient

Comparison of RNN variants







dprogrammer.org

Time series prediction with RNNs

- Formulate as a regression problem
  - Input: time series
  - Output: shifted time series
- Sliding window approach
  - Limiting data used for predictions

- General:
  - Scaling: normalizing is necessary when using neural networks (activation functions tanh, sigmoid...)
  - Subtraction of known signals recommended (trend, seasonal signals,…)
  - Selection of hyperparameters (network topology) with validation dataset or cross-validation

Machine learning strategies in geodetic applications

Example: prediction of ionospheric parameters

- Ionospheric corrections are needed for single-frequency positioning
- Predictions are needed in real-time scenarios to overcome latency

Ensemble Machine Learning of Random Forest, AdaBoost and XGBoost for Vertical Total Electron Content Forecasting

by **Randa Natras** [1,*], **Benedikt Soja** [2] and **Michael Schmidt** [1]

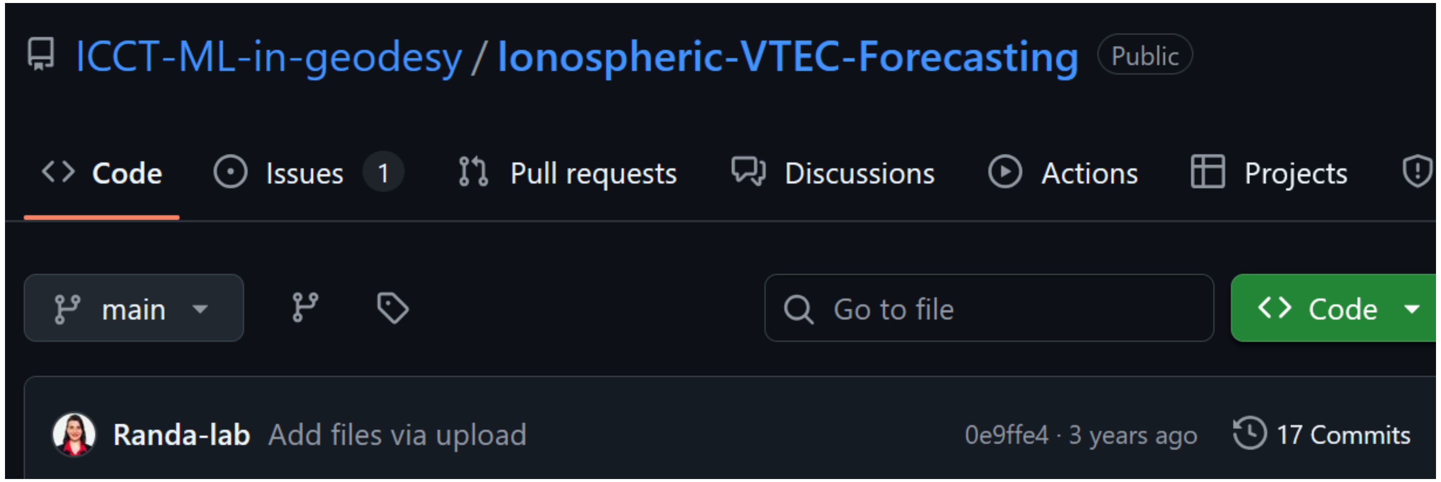
1 Deutsches Geodätisches Forschungsinstitut (DGFI-TUM), TUM School of Engineering and Design, Technical University of Munich, 80333 Munich, Germany

2 Institute of Geodesy and Photogrammetry, ETH Zurich, 8093 Zurich, Switzerland

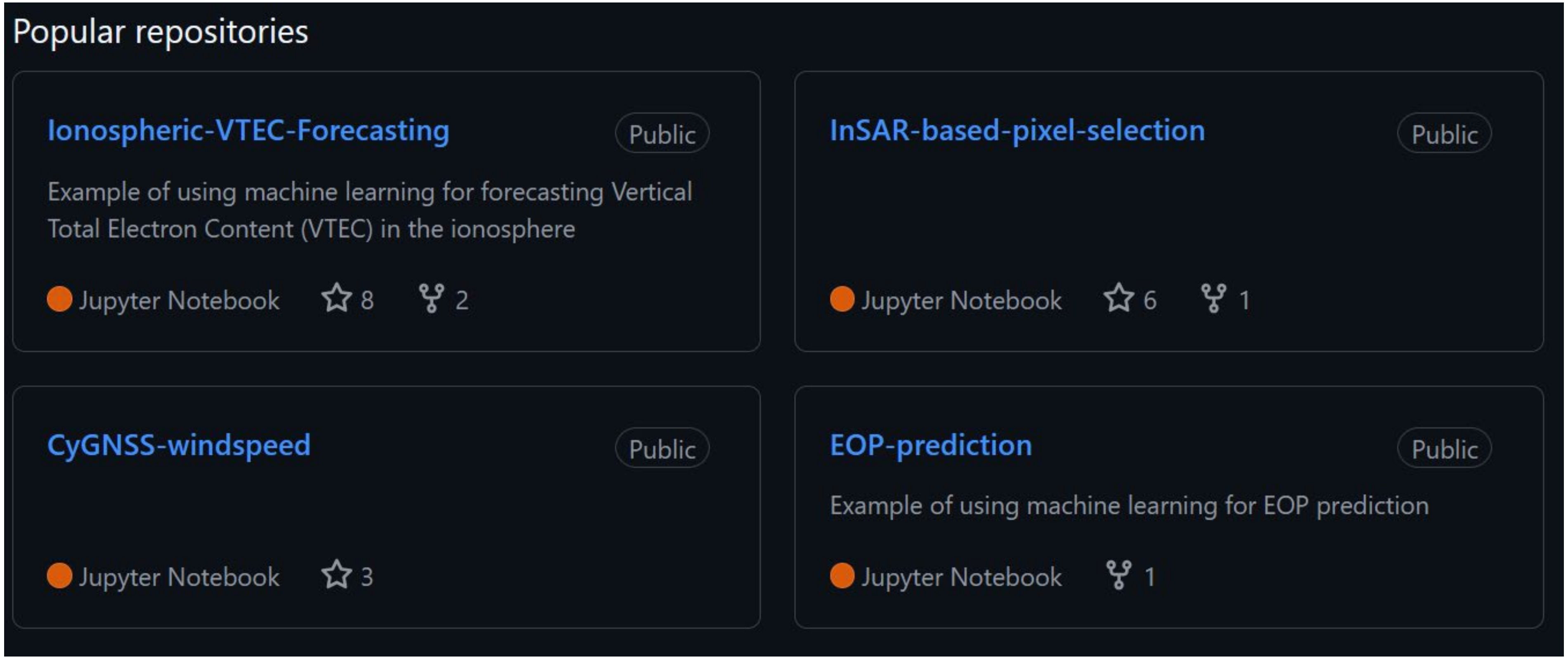
* Author to whom correspondence should be addressed.

Remote Sens. **2022**, *14*(15), 3547; **https://doi.org/10.3390/rs14153547**

Example of ionospheric prediction as Jupyter Notebook



Further toy examples of machine learning in geodesy



Problem definition

- Which type of problem?
  - Supervised, unsupervised, …
  - Regression, classification, …

- Which data?
  - Time series, images, …
  - Discrete, continuous
  - Gridded, irregular, sparse
  - Quantity
  - Quality
  - Distribution of samples
  - Metadata

Guidance on algorithm selection

- Classical machine learning algorithms

Aspect	**SVM**	**DT**	**RF**	**XGBoost**
Approach	Maximum margin classifier	Recursive partitioning	Ensemble of decision trees	Ensemble of boosted trees
Type	classification, regression	classification, regression	classification, regression	classification, regression
Interpretability	Low (unless using linear kernel)	High	Interpretable tree-based methods (e.g., [29]	Low (ensemble, complex optimization)
Overfitting	Sensitive (with high complexity data)	Prone to overfitting	Reduced overfitting	Reduced overfitting (with regularization)
Performance	Moderate for complex problems	Prone to poor performance on large datasets	Strong (good balance of accuracy and speed)	Very high (often superior performance)
Data Requirements	high-dimensional data	Can handle both numerical and categorical data	Can handle large and complex datasets	Handles large datasets efficiently
Best Use Case	High-dimensional, sparse datasets	Small to medium-sized datasets, clear decision rules	Large datasets, requiring robust performance	Large, complex datasets, high accuracy needed

Kaselimi & Soja, in prep.

Guidance on algorithm selection

- Deep learning algorithms

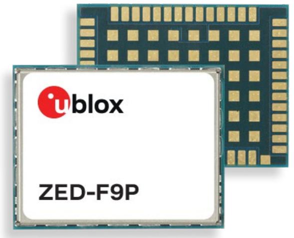
Aspect	Convolutional Neural Networks	Recurrent Neural Networks, including LSTMs	UNet	Transformers	Graph Neural Networks
Approach	Convolutional layers capture spatial features	Sequential processing through recurrent connections	Encoder-decoder with skip connections	Attention mechanism with self-attention	Captures relationships in graph-structured data
Type	Supervised (classification, regression)	Supervised (classification, regression)	Supervised (segmentation, reconstruction)	Supervised (classification, regression)	Supervised (node, edge, graph-level tasks)
Interpretability	Medium (spatial feature maps)	Low (difficult to interpret long sequences)	Medium (interpretable for segmentation)	Low (complex attention weights)	Medium (depends on graph size and structure)
Overfitting	Sensitive (with small datasets)	Prone (for long sequences)	Moderate (depends on skip connections)	Reduced (requires large data)	Reduced (with regularization and dropout)
Performance	High for spatial data (e.g., images)	High for sequential data (e.g., time-series, NLP)	High for image segmentation tasks	Very high (state-of-the-art for sequences)	High for graph-based tasks
Data Requirements	Moderate to large datasets	Moderate to large datasets	Moderate to large datasets	Large datasets (benefits from pretraining)	Small to large datasets
Best Use Case	Image-based geodesy tasks, e.g., deformation monitoring from satellite imagery, land cover classification.	Time-series analysis in geodesy, e.g., GPS time-series for crustal deformation, earthquake precursors.	Segmentation in geodesy, e.g., detecting boundaries in remote sensing imagery, glacier delineation.	Long-sequence tasks in geodesy, e.g., geodetic time-series modeling, language-based metadata classification.	Analyzing geospatial networks, e.g., road network analysis, tectonic plate interaction modeling.

Kaselimi & Soja, in prep.

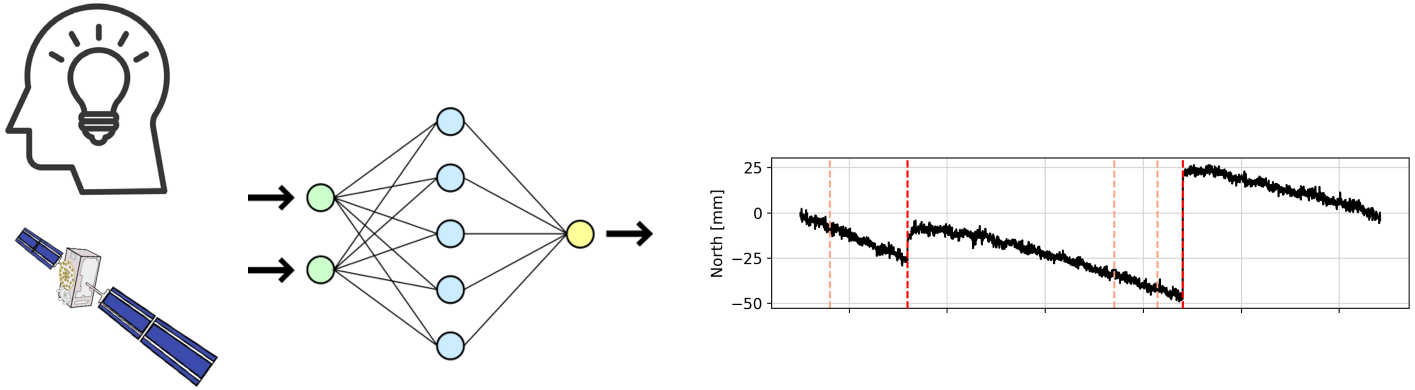
Perspectives for Machine Learning in Geodesy

- ML will probably have a place in all areas of geodesy – but not for all tasks!
  - Parallels with neighboring fields, e.g. seismology or remote sensing

- Difficult to properly exploit the available geodetic data
  - Efficiency of deep learning algorithms could become a necessity
  - Accelerated by low-cost sensors and smallsats




- Domain knowledge will always be essential



Challenges for Machine Learning in Geodesy

- Technical problems: data selection & preprocessing, algorithm selection & tuning, training process, …
  - Enhancing data science literacy in education
  - Recruiting data science expertise
  - Setting up benchmarks/evaluation protocols (e.g., hackathons, EOP PCC)

- Lack of trust & interpretability (“black box”)
  - Explainable learning
    - Uncertainty quantification
    - Feature importance
  - Physics-based learning

Conclusions

- Machine learning shows great promise in data-rich fields – including geodesy!
- Wide variety of ML problems and algorithms
  - Select the right one for your application!
  - Benefit from open-source ML libraries and examples
- Lots of new opportunities – experiment and have fun!

Benedikt Soja
soja@ethz.ch

ETH Zurich
Chair of Space Geodesy
Institute of Geodesy and Photogrammetry
Zurich, Switzerland
https://space.igp.ethz.ch

Thanks for your attention!