CS-E4740 - Federated Learning

Welcome and Introduction

Assoc. Prof. Alexander Jung

Spring 2025

Playlist

Glossary

Course Site

At a Glance

- 5 credits, fully online - no mandatory attendance.
- Six modules with lecture, exercise and assignment.
- Instead of assignments, students can review papers.
- We will also have a student project with peer review.
- Potential extension to 10 credits decided case-by-case.

Table of Contents

Table of Contents

Prerequisites

- **Linear Algebra.** Vectors $\mathbf{w} \in \mathbb{R}^n$, matrices $\mathbf{Q} \in \mathbb{R}^{n \times n}$, norms $\|\mathbf{w}\|_2$.
- **Multivariable Calculus.** Smooth functions $f(\mathbf{w})$ and their gradient $\nabla f(\mathbf{w})$.
- **Basic Machine Learning.** Empirical risk minimization $\min_{h \in \mathcal{H}} (1/m) \sum_{r=1}^{m} L\left(\left(\mathbf{x}^{(r)}, y^{(r)}\right), h\right)$.
- **Python.** Basic coding skills and familiarity with libraries `numpy` and `scikit-learn`

Learning Goals

After completing this course, you can

- model FL applications using network models
- formulate FL as an optimization problem
- design FL algorithms via distributed optimization
- build trustworthy FL systems.

Table of Contents

Positioning of CS-E4740 in ML Curriculum

In what follows, we briefly explain how CS-E4740 relates to selected courses at Aalto University and University of Helsinki.

Related Courses - Bare Necessities

- **MS-A0001 - Matrix Algebra.** Introduction to linear algebra in $\mathbb{R}^d$. We will use $\mathbb{R}^d$ as a mathematical model for FL.
- **CS-C3240 - Machine Learning.** Teaches basic techniques for training a single ML model on a given dataset. FL extends this centralized setting to networks of devices, each having access to a local dataset and a local (personalized) model.
- **Data Analysis with Python.** Teaches how to implement basic ML methods in Python. Our course assignments require to implement (parts of) FL algorithms in Python.

Related Courses - Nice to Have

- **MS-C2105 - Introduction to Optimization.** Teaches basic concepts for the design and analysis of optimization methods. Our course formulates FL as an optimization problem. FL algorithms are obtained, in turn, by applying optimization methods to solve this problem.

- **ELEC-E5424 - Convex Optimization.** Teaches advanced tools (such as convergence analysis of gradient methods) for the study and design of FL algorithms.

Related Courses - Follow Up

- **ELEC-E7120 - Wireless Systems.** Discusses the fundamentals of radio communications which can be used to implement FL algorithms.
- **ELEC-E8102 - Distributed and Intelligent Automation Systems.** Discusses automation systems consisting of interconnected sensors and actuators. We can use FL to train predictive models used by these devices.

Table of Contents

Six Modules

- **ML Refresher.** Model training, validation, and regularization.
- **FL Networks.** Use graphs to model FL networks.
- **FL Design Principle.** Formulate FL as optimization over a graph.
- **FL Algorithms.** FL via distributed optimization methods.
- **FL Flavours.** Clustered, vertical, horizontal and more.
- **Trustworthy FL.** Explainability, robustness, and privacy-protection.

Each module (**M**) consists of a lecture (**L**) and assignment (**A**)

Graded Activities

- **Assignments.** Implement concepts in Python.
- **Project.** Study FL application of your wish.
- Instead of (in addition to) assignments you can
  - **Review a paper** from a curated list.
  - Interpret it using the concepts taught in the lectures.
  - Present your review in a slide talk.
  - Submit slides and recorded talk (max. 10 minutes).

Paper List: 

Grading

The grading is based on the points collected via

- 6 assignments, max. $6 \cdot 7 = 42$ points (Assgts)
- peer grading of projects, max. 7 points (PG)
- final project report, max. 51 points $(\mathrm{Project})$
- paper review, max. 42 points (Paper)

Total number of points $= \max\big(\mathrm{Assgts}, \mathrm{Paper}\big) + \mathrm{Project} + \mathrm{PG}$

From Points to Grade

- **Grade 1** for 50-59 points.
- **Grade 2** for 60-69 points.
- **Grade 3** for 70-79 points.
- **Grade 4** for 80-89 points.
- **Top grade 5** for at least 90 points.

Assignments

Six assignments **A1**,. . .,**A6**, one for each module **M1**,. . .,**M6**, require implementing concepts in Python.

- (**A1**) ML model training, validation and regularization.
- (**A2**) From ML to FL.
- (**A3**) FL as optimization over networks.
- (**A4**) FL algorithms as message passing.
- (**A5**) Clustered and vertical FL.
- (**A6**) Privacy attacks and subjective explainability.

Student Project

- Choose an FL application of your choice.
- Design and study FL algorithms from the course.
- Write a project report (we will provide template).
- Submit your report by the end of April.
- Peer-review other's reports until 15-May.
- Submit revised report and response letter by end of May.

Schedule

- (**L**) Each Mon. at 16:15, starting 24-Feb.
- (**A**) Presentation each Wed. 16:15, starting 26-Feb.
- **A1**, **A2** must be submitted by 17-Mar-2025
- **A3**, **A4** by 31-Mar-2025
- **A5**, **A6** by 14-April-2025
- First project report submission by 30-April-2025
- Peer grading until 15-May-2025
- Final project submission by 31-May-2025

Ground Rules

As a student following this course, you must act according to the Aalto University Code of Conduct (see here).

Two main ground rules for this course are . . .

Rule I - Be Honest

- This course includes a lot of independent work:
  - completing assignments,
  - preparing student project report,
  - peer grading others projects,
  - completing paper review.
- Do not steal (plagiarize) others work!
- You must indicate (cite) any sources used!
- Randomly selected students need to explain their work.

Rule II - Be Respectful

My personal wish is that this course provides a safe space for an enjoyable learning experience.

Any form of disrespectful behaviour, including any course-related communication platforms, will be sanctioned rigorously (including reporting to university authorities).

Table of Contents

What is Federated Learning?

Federated Learning (FL) trains machine learning (ML) models in a distributed fashion over a network of devices.

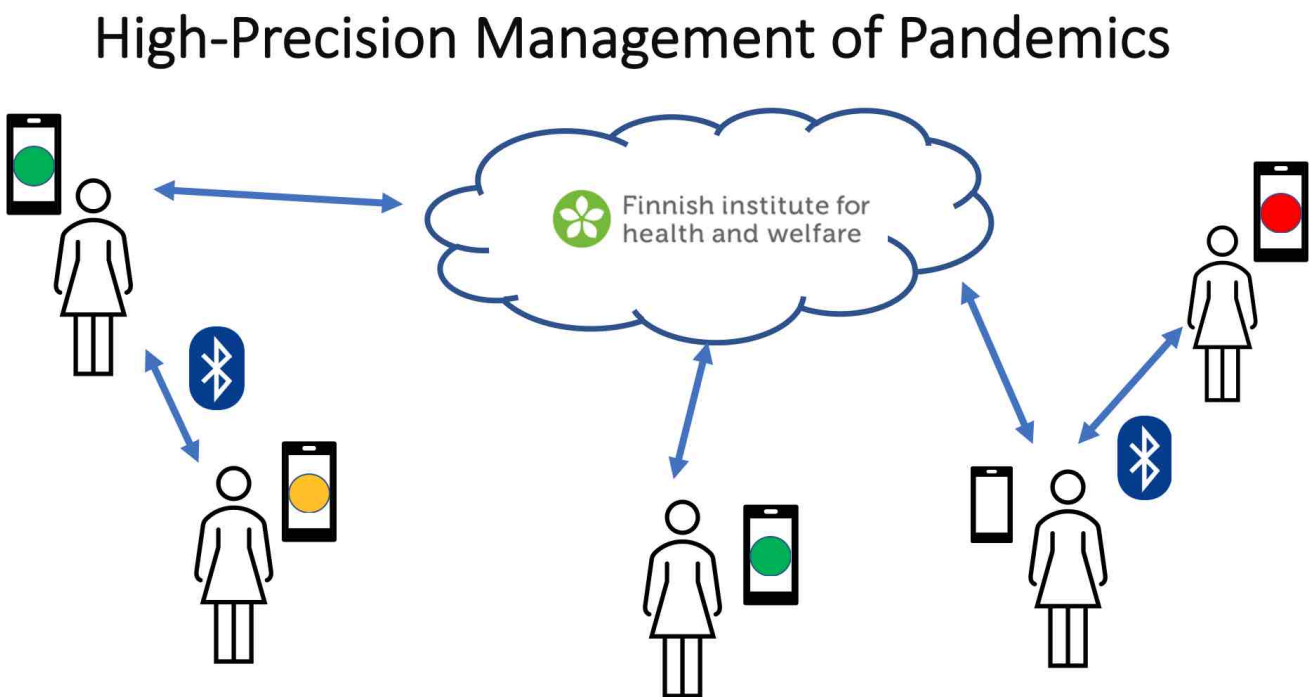
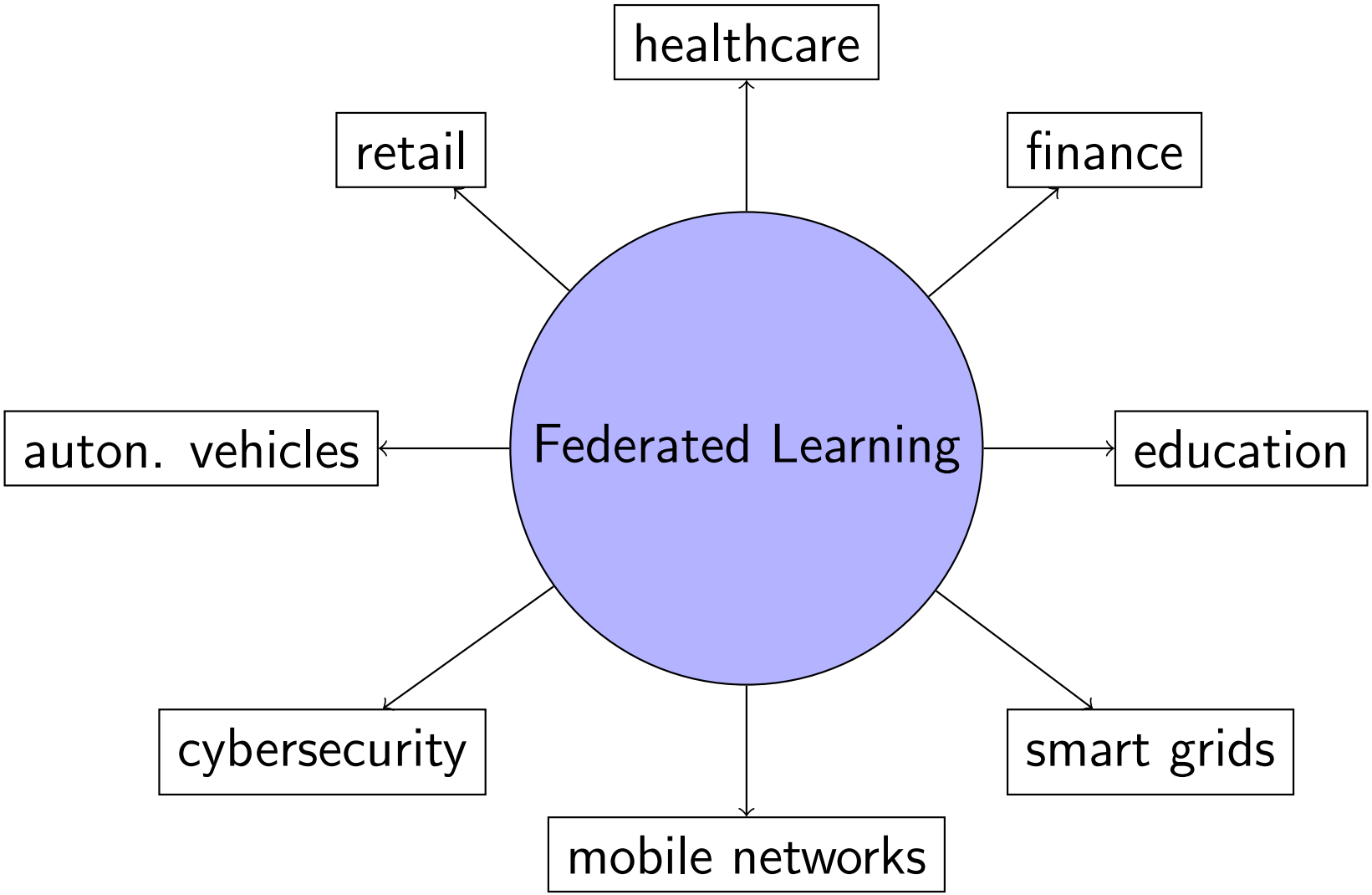


Figure: A Federated Learning system for prediction of infections. Smartphones train personalized models based on their observations (audio recordings of a cough) as well as public health-care data.

Key Characteristics of FL

- No centralized data collection (robustness).
- Each device trains personalized model (high-precision).
- Share information/compute among devices (scalability).
- No raw data is shared (privacy-friendly).

FL Applications



FL in Healthcare

- Turn smartphones into a health-care advisors.
- Smartphone app uses FL to train personalized model.
- Combine personal data with public health-care data.

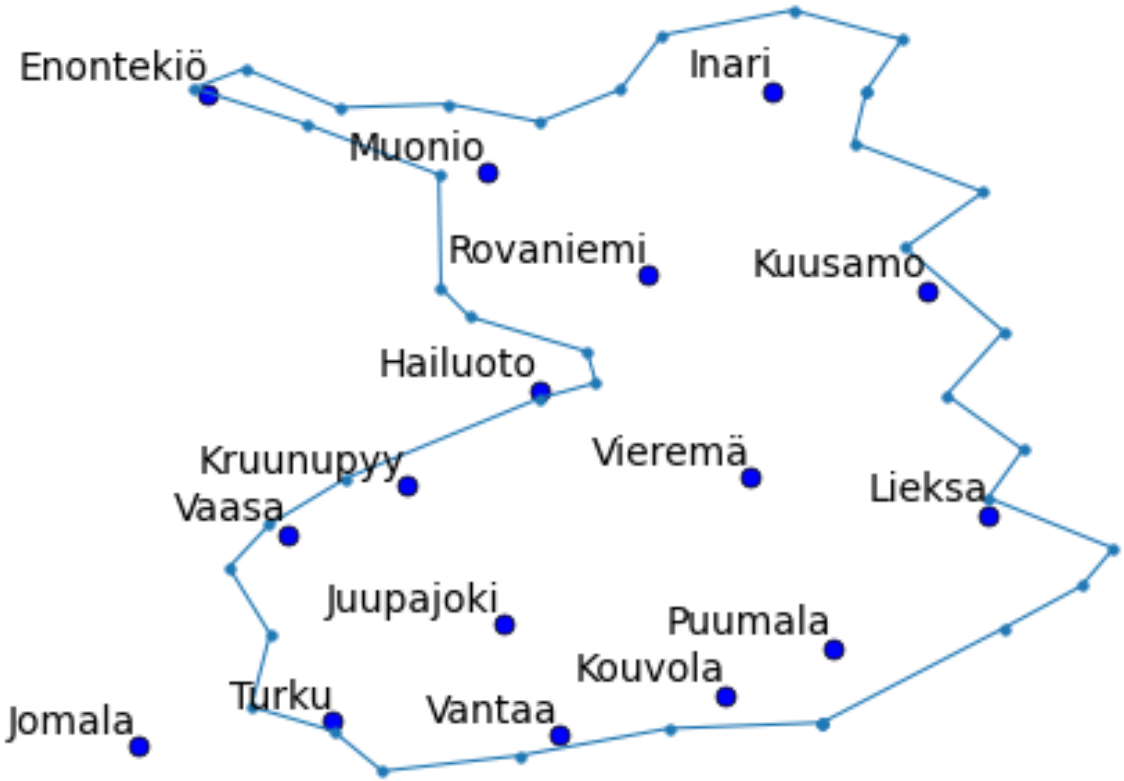
Key Reference: Rieke, N., et al. *The future of digital health with federated learning.* Nature Medicine, 2020.

FL in Finance

FL can help financial institutions to improve

- **Fraud detection.** N. F. Aurna, et.al., "Federated Learning-Based Credit Card Fraud Detection: Performance Analysis with Sampling Methods and Deep Learning Algorithms," 2023,
- **Risk assessment.** W. Li, et.al., "Personal Credit Evaluation Model Based on Federated Learning," 2024

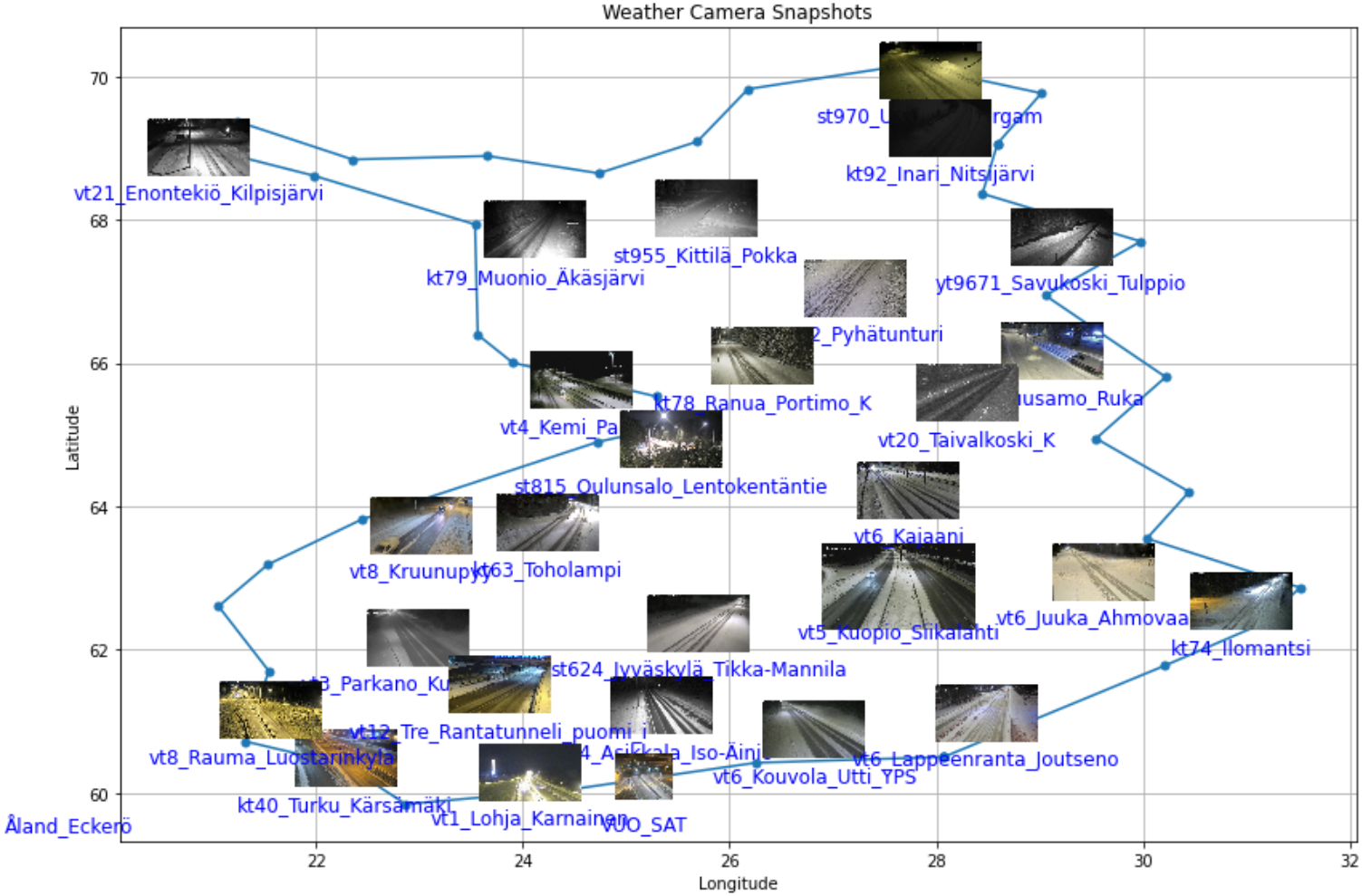
FL at the Finnish Meteorological Institute (FMI)



Train a separate model for each FMI weather station

Python script for reproducing the Fig.:

FL for Finnish Road Safety



Train local model for each camera operated by FinTraffic

Python script for reproducing the Fig.:

The Internet of Things (IoT) is Growing

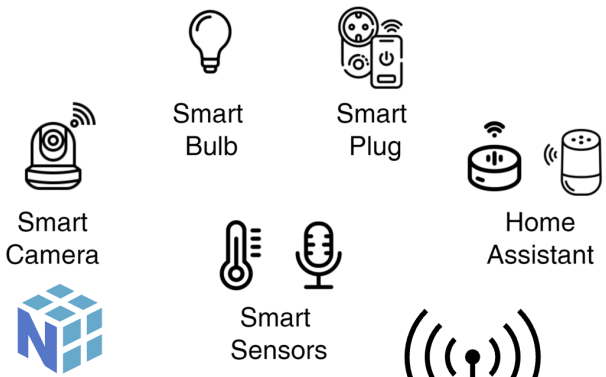
IoT connections (billion)

IoT	2023	2029	CAGR
Wide-area IoT	3.6	7.2	12%
Cellular IoT	3.4	6.7	12%
Short-range IoT	12.1	31.6	17%
Total	**15.7**	**38.8**	**16%**

Note: Based on rounded figures. Cellular IoT figures are also included in the figures for wide-area IoT.

Figure: Some IoT statistics from https://www.ericsson.com/en/reports-and-papers/mobility-report/dataforecasts/iot-connections-outlook.

The IoT - A Global FL System



From ML to FL

- Basic ML: Train a single model $\mathcal{H}$ by minimizing average loss on a single dataset
- FL: Train a separate model $\mathcal{H}^{(i)}$ for each device i using decentralized data

ML with Python

```
X, y = read_data()
model = SGDRegressor()
model.fit(X, y)
```

FL with Python

IP: 192.168.0.1

```
X, y = read_data()
model = SGDRegressor()
model.fit(X, y)
```

IP: 192.168.0.2

```
X, y = read_data()
model = LinearRegression()
model.fit(X, y)
```

IP: 192.168.0.3

```
X, y = read_data()
model = DecisionTree()
model.fit(X, y)
```

FL $=$ ML $\otimes \ldots \otimes$ ML

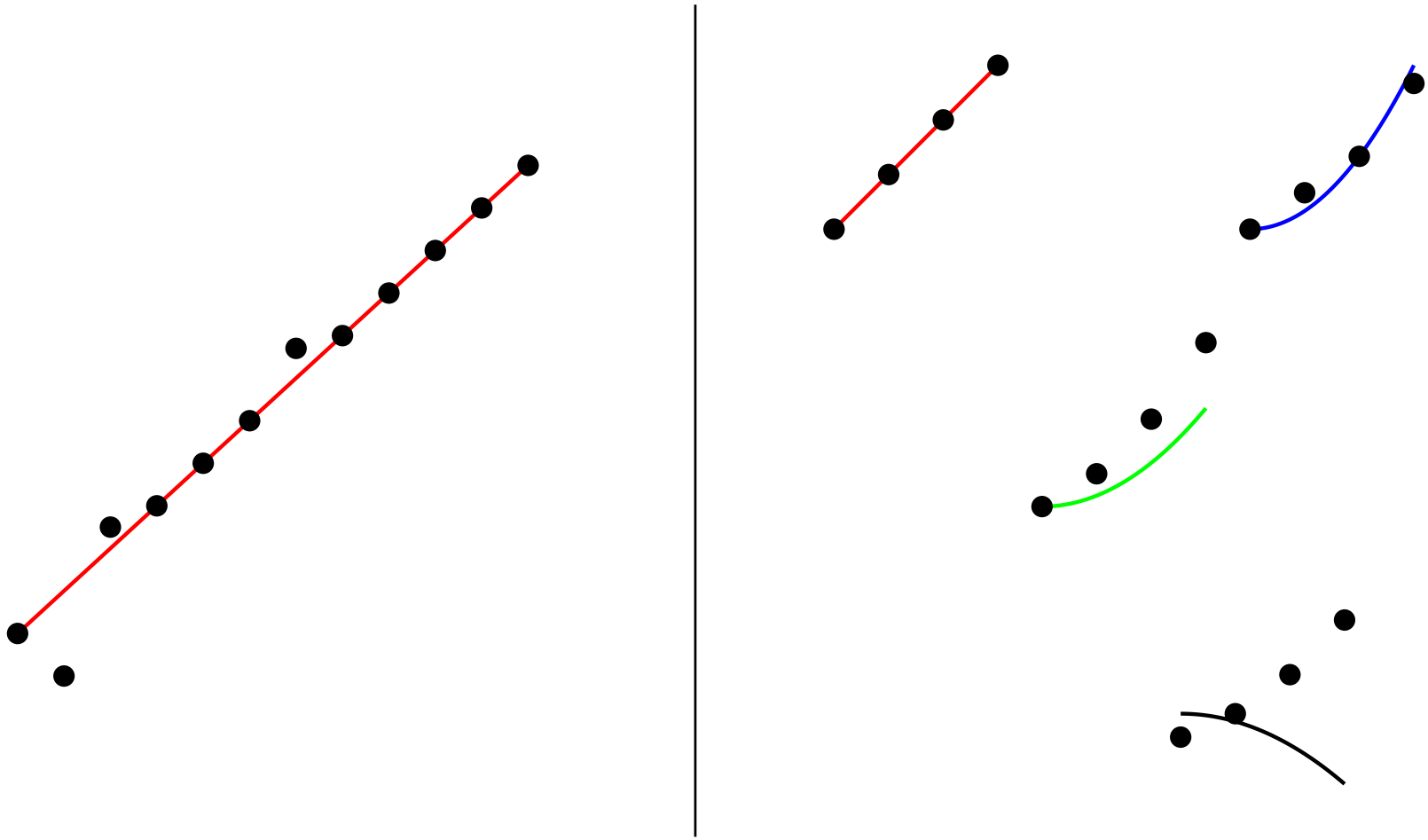
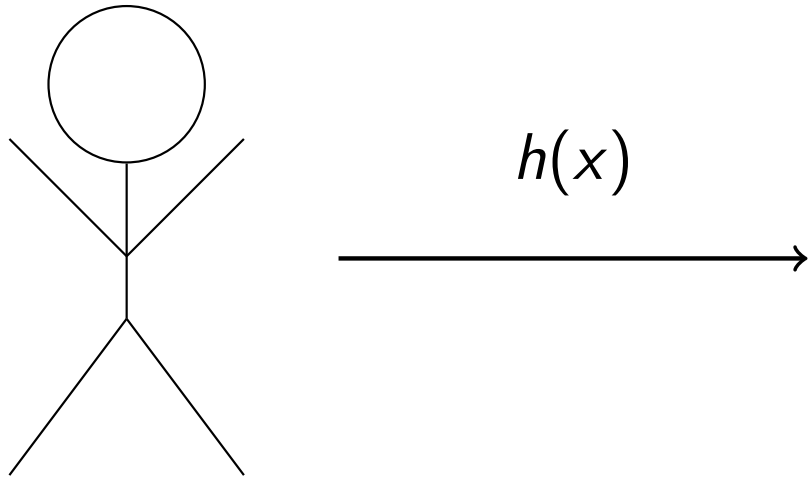


Figure: Left: A basic ML method uses a single dataset to train a single model. Right: FL methods train personalized models from decentralized datasets.

Table of Contents

The Right Song Can Save a Day



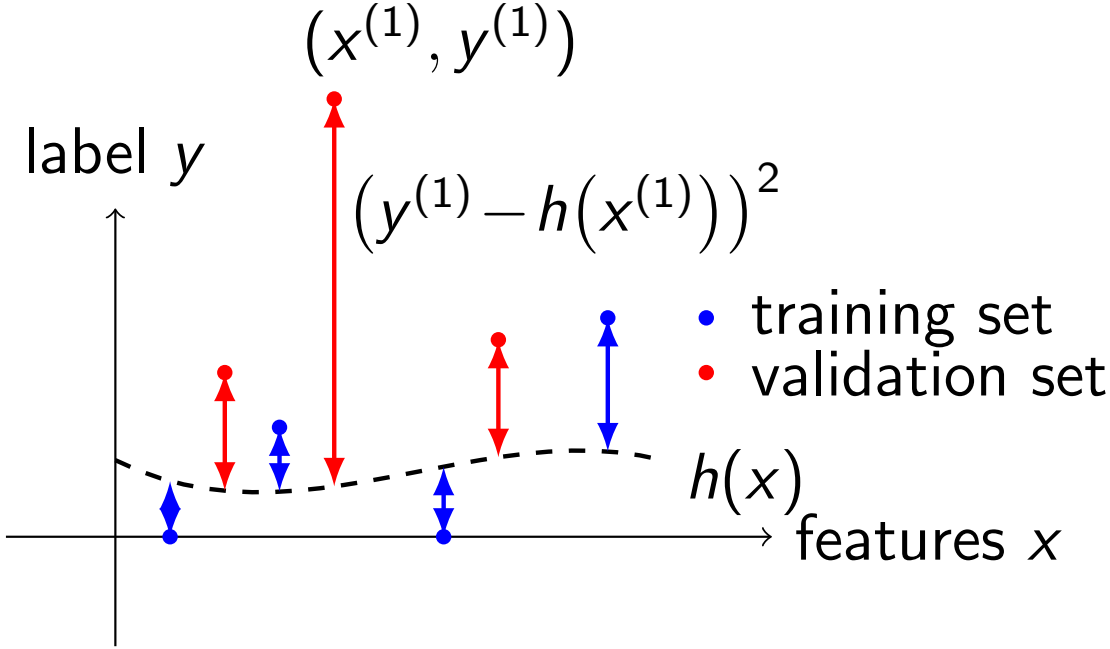
user: Alex Jung
mood: tired
evening

Bohemian Rhapsody

How do we get a good hypothesis map $h(x)$?

Wang, M., Wu, J., Yan, H. (2023). "Effect of music therapy on older adults with depression: A systematic review and meta-analysis." *Complementary Therapies in Clinical Practice* https://doi.org/10.1016/j.ctcp.2023.101809

Empirical Risk Minimization (ERM)



ERM learns h from model $\mathcal{H}$ by min. average loss (empirical risk),

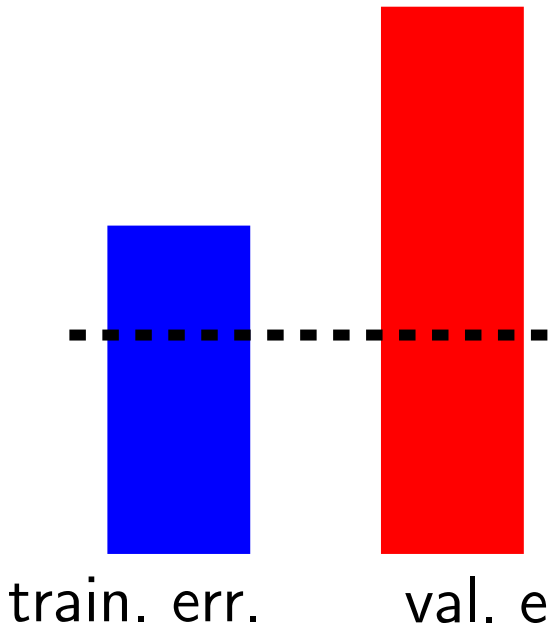
$$\min_{h \in \mathcal{H}} \frac{1}{m} \sum_{r=1}^{m} L\left(\left(\mathbf{x}, y\right), h\right)$$

Different choices for $\mathcal{H}$ and loss L yield different ML methods.

---

see Chapters 3,4 of AJ, "Machine Learning: The Basics," Springer, 2022. https://mlbook.cs.aalto.fi

Applied ML - Trial and Error

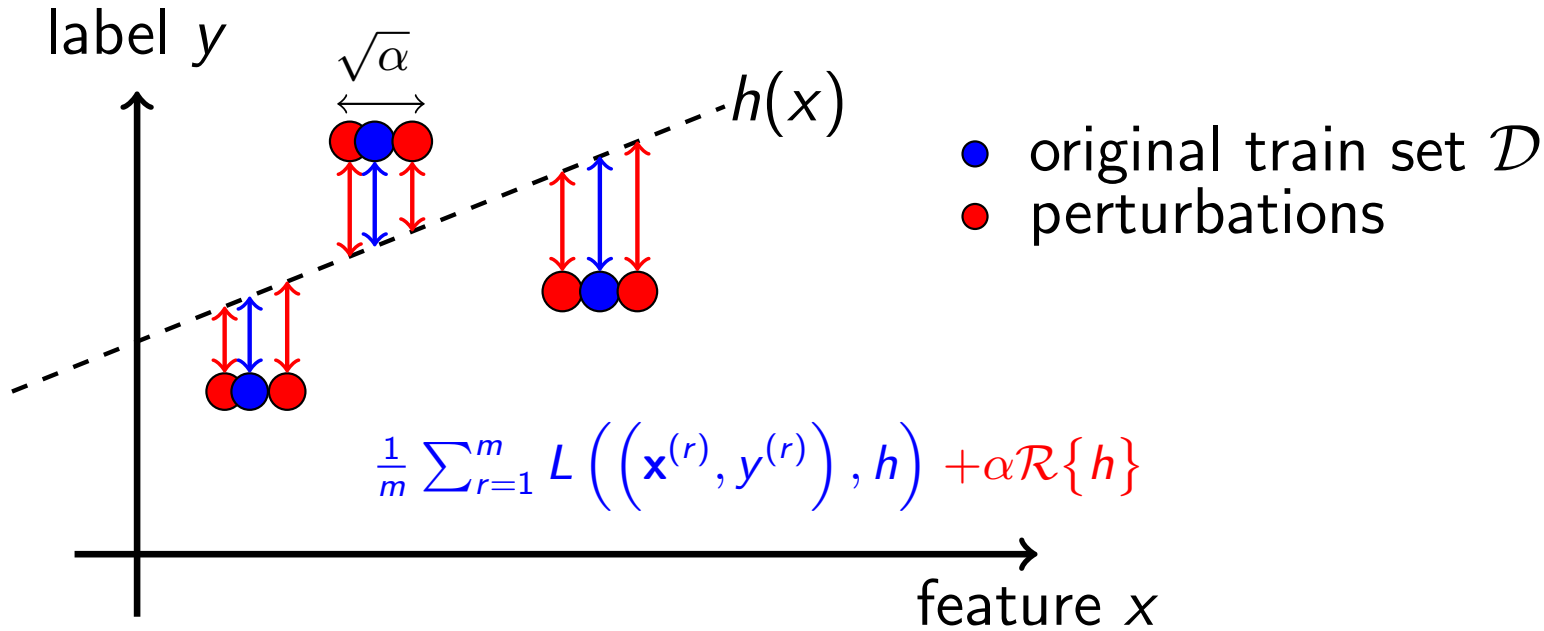


Diagnose and select ML models by train. and val. err; if both are on same level as baseline (or benchmark) you are done!

---

see Chapter 6 of AJ, "Machine Learning: The Basics," Springer, 2022. https://mlbook.cs.aalto.fi

Applied ML - Regularization



Start with large $\mathcal{H}$, then reduce its effective size either by

- **data augmentation**, e.g., $\mathbf{x} \mapsto \mathbf{x} + \mathcal{N}(0, \alpha)$, or
- adding **penalty** to loss function, e.g., $\ldots + \alpha \left\|\mathbf{w}\right\|_2^2$,
- or **constraints on model parameters**, e.g., $\left\|\mathbf{w}\right\|_2 \leq 1$.

---

see Chapter 7 of AJ, "Machine Learning: The Basics," Springer, 2022.
https://mlbook.cs.aalto.fi

What's Next?

The next module introduces FL networks as our main mathematical model for FL applications.

Later modules use FL networks for the design and analysis of FL systems.