

CS-E4740 - Federated Learning

# L1 - From ML to FL

Assoc. Prof. Alexander Jung

Spring 2026

**Calendar**



**Glossary**



**Book**



**GitHub**



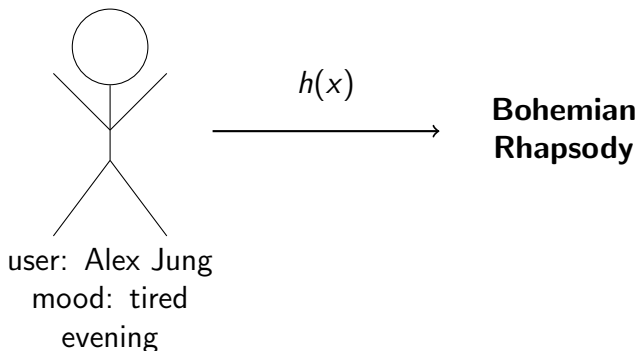
# Table of Contents

Machine learning (ML) Basics

From ML to federated learning (FL)

Federated Learning Networks (FL networks)

# The Right Song Can Save the Day



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

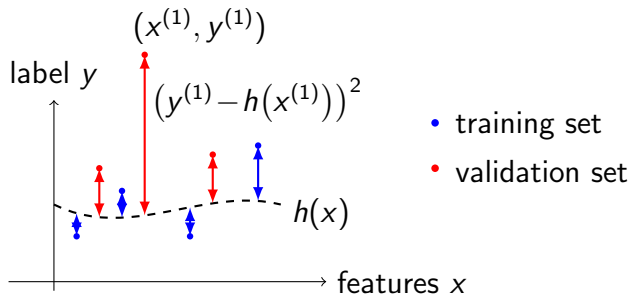
---

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)



Learn  $h \in \mathcal{H}$  by min. average loss (empirical risk),

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

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>

# ML with Python

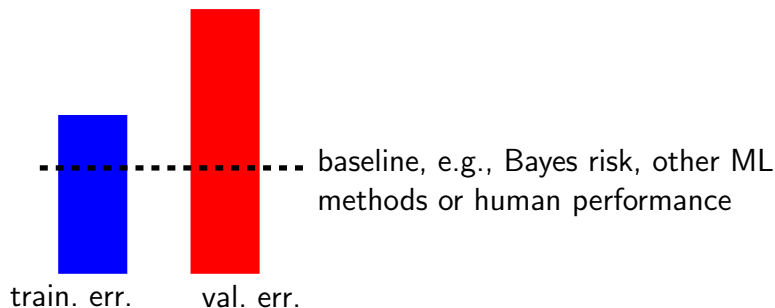
```
X, y = read_data()

# split data
Xtr, Xval, ytr, yval = train_test_split(X, y)

# train model
model = SGDRegressor()
model.fit(Xtr, ytr)

# compute errors
train_err = mean_squared_error(ytr, model.predict(Xtr))
val_err    = mean_squared_error(yval, model.predict(Xval))
```

# Applied ML - Diagnosis

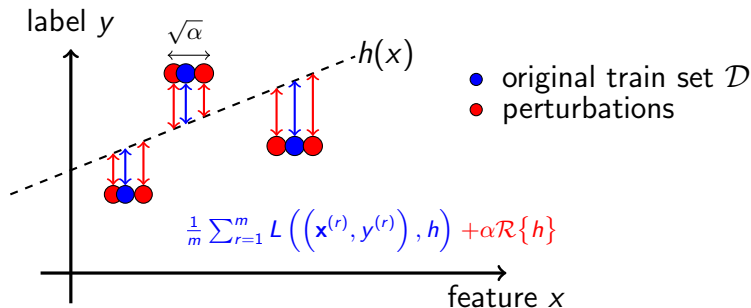


compare training error with validation error and a baseline.

---

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 shrink it via (combinations of)

- ▶ data augmentation, e.g.,  $\mathbf{x} \mapsto \mathbf{x} + \mathcal{N}(0, \alpha)$ ,
- ▶ adding penalty term to loss function, e.g.,  $\dots + \alpha \|\mathbf{w}\|_2^2$ ,
- ▶ **constraining** model parameters, e.g.,  $\|\mathbf{w}\|_2 \leq 1$ .

---

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

# Table of Contents

ML Basics

From ML to FL

FL networks



# From ML to FL

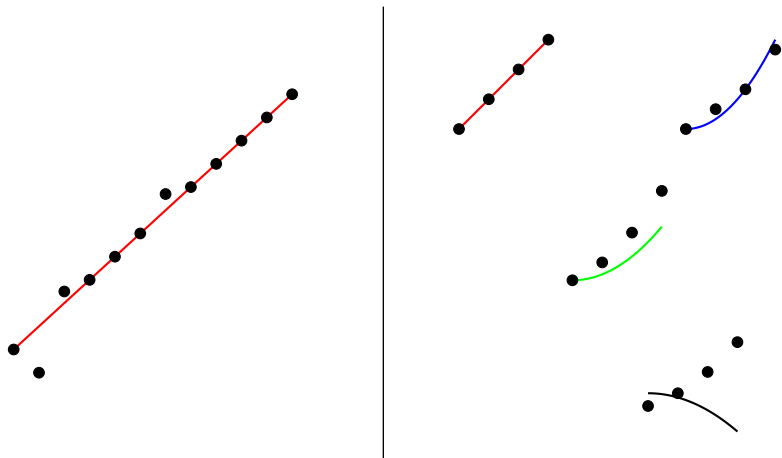
**Basic ML.** Train a single model  $\mathcal{H}$  by minimizing average loss on a single dataset

**FL.** Train several models  $\mathcal{H}^{(i)}$  using interconnected device.

We use the term device broadly. It is anything that can

- ▶ access data,
- ▶ train a model, and
- ▶ communicate with other devices.

# From ML to FL



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

# ML with Python

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

# FL with Python

IP: 192.168.0.1

```
model = SGDRegressor()  
y_hat =recv_preds(192.168.0.3)  
X, y = read_data()  
Xa,ya=augment_data(X, y, y_hat)  
model.fit(Xa,ya)
```

IP: 192.168.0.2

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

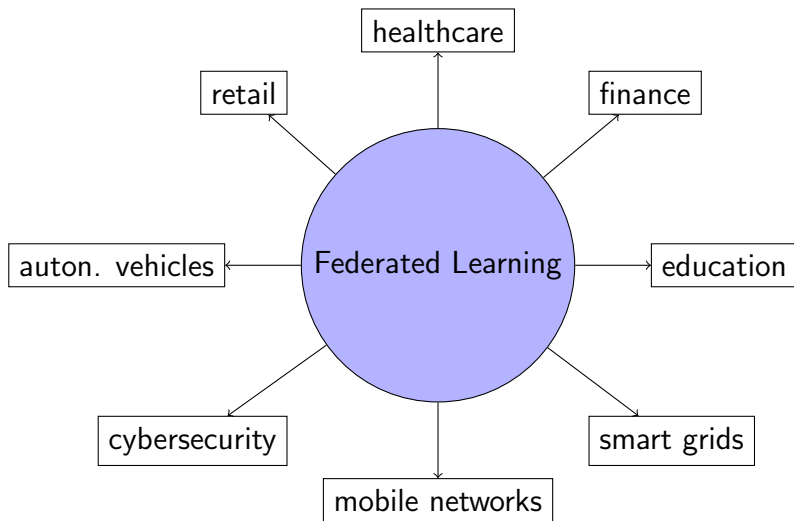
IP: 192.168.0.3

```
model=DecisionTree()  
y_hat =recv_preds(192.168.0.2)  
X, y = read_data()  
Xa,ya=augment_data(X, y, y_hat)  
model.fit(Xa,ya)
```

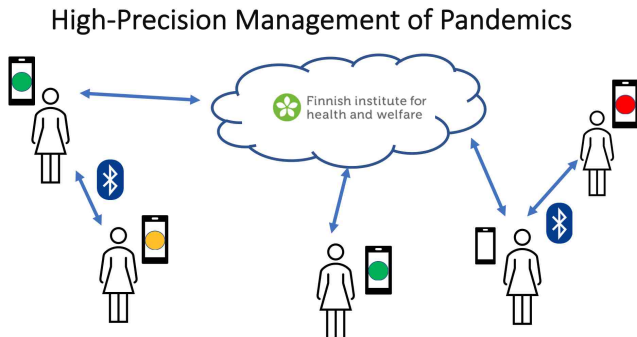
# Key Characteristics of FL

- ▶ No centralized data collection (no single point of failure)).
- ▶ Each device trains a tailored model (high-precision).
- ▶ Scalability: more devices yield more compute and data.
- ▶ No raw data is shared (privacy-friendly).

# FL Applications



# FL for Pandemics



**Figure:** A hypothetical FL system for pandemic forecasting. Smartphones train personalized models based on their observations (e.g., audio recordings of coughing) as well as public health-care data.

# FL in Healthcare

- ▶ Turn smartphone into personal health-care advisor.
- ▶ 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.

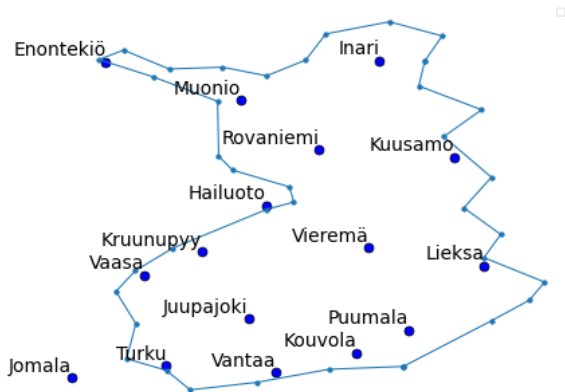


# FL in Finance

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 FMI

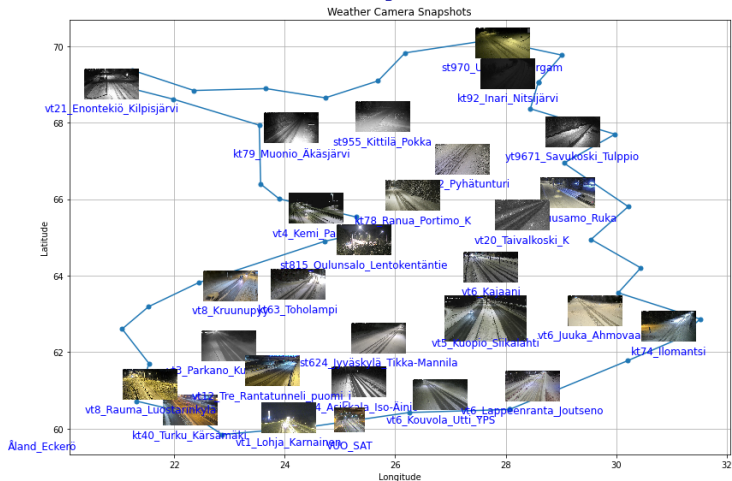


Train a separate model for each Finnish Meteorological Institute (FMI) weather station

Python script for reproducing the Fig.:



# FL for Finnish Road Safety



Train separate model for each camera operated by FinTraffic

Python script for reproducing the Fig.:



# The Internet of Things (IoT) is Growing

## IoT connections (billion)

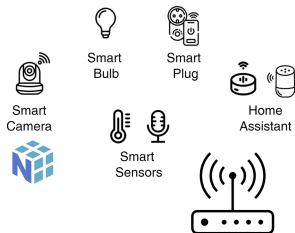
IoT	2023	2029	CAGR
Wide-area IoT	3.6	7.2	12%
Cellular IoT	3.4	6.7	12%
Short-range IoT	12.1	31.6	17%
<b>Total</b>	<b>15.7</b>	<b>38.8</b>	<b>16%</b>

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

Figure: Some IoT statistics from



# The IoT - A Global FL System



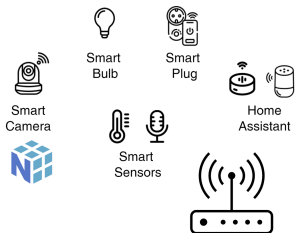
# Table of Contents

ML Basics

From ML to FL

FL networks

# A (“Real-World”) FL System



# Abstracting Away System Details

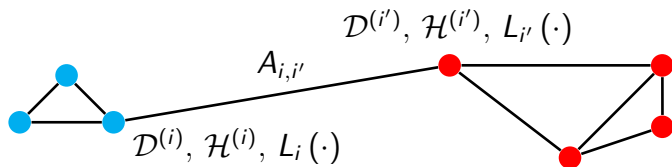
to reason about an FL system, we deliberately ignore many implementation details:

- ▶ physical properties of communication links (latency, bandwidth)
- ▶ communication protocols and message formats
- ▶ hardware and operating systems of devices
- ▶ software stacks and scientific computing libraries

**Goal:** isolate the *essential structure* of a FL system needed to analyze the overall behaviour.

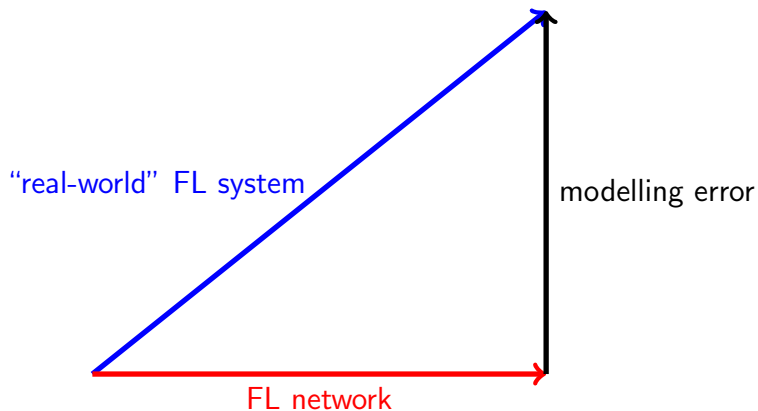


# The FL network as an Abstraction



- ▶ an FL network is an undirected graph with nodes  $i=1, \dots, n$
- ▶ edge  $\{i, i'\}$  with weight  $A_{i,i'} > 0$  encodes collaboration
- ▶ each node  $i$  holds local dataset  $\mathcal{D}^{(i)}$  and trains model  $\mathcal{H}^{(i)}$
- ▶ a local dataset induces a local loss function  $L_i(\cdot)$

# FL network is an Approximation



# A Precise Definition

An FL network consists of:

- ▶ a finite set of **nodes**, denoted as  $\mathcal{V} := \{1, \dots, n\}$
- ▶ a **local model**  $\mathcal{H}^{(i)}$  at each node  $i \in \mathcal{V}$
- ▶ a **local loss function**  $L_i(\cdot)$  at each node  $i \in \mathcal{V}$
- ▶ a set of undirected **edges**, denoted as  $\mathcal{E}$
- ▶ a positive **edge weight**  $A_{i,i'} > 0$  for each edge  $\{i, i'\} \in \mathcal{E}$

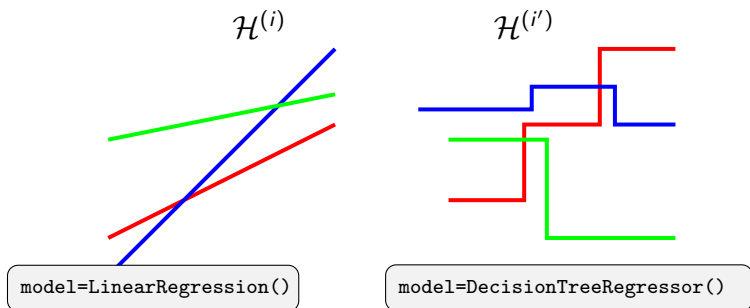
Thus, a FL network is a weighted undirected graph with a model and loss function attached to each node.

# Nodes of an FL network

- ▶ consider an FL system with finite number  $n$  of devices
- ▶ we index devices as  $i = 1, \dots, n$
- ▶ indices form the set of nodes  $\mathcal{V}$  in an FL network.
- ▶ node  $i \in \mathcal{V}$  **represents** a physical device.
- ▶ we use “device  $i$ ” and “node  $i$ ” interchangeably.

# Local models

- ▶ consider an FL system with devices  $i = 1, \dots, n$ .
- ▶ each device trains local (personal) model  $\mathcal{H}^{(i)}$ .
- ▶ devices might use (very) different local models.
- ▶ we use local model parameters  $\mathbf{w}^{(i)}$  for parametric  $\mathcal{H}^{(i)}$ .



# Local Loss functions

- ▶ consider device  $i$ , training its local model  $\mathcal{H}^{(i)}$ .
- ▶ *to train a model* is to learn a useful hypothesis  $h^{(i)} \in \mathcal{H}^{(i)}$ .
- ▶ measure usefulness of  $h^{(i)}$  by a local loss function

$$L_i(\cdot) : \mathcal{H}^{(i)} \rightarrow \mathbb{R} : h^{(i)} \mapsto L_i(h^{(i)})$$

- ▶ different devices can use different loss functions.

## Local Loss functions - ctd.

- ▶ FL methods use different constructions of loss functions
- ▶ for parametric models  $\mathcal{H}^{(i)}$ , with model parameters  $\mathbf{w}^{(i)} \in \mathbb{R}^d$ ,

$$L_i(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R} : \mathbf{w}^{(i)} \mapsto L_i(\mathbf{w}^{(i)})$$

- ▶ can use average loss on local dataset

$$L_i(\mathbf{w}^{(i)}) := \frac{1}{m_i} \sum_{r=1}^{m_i} \left( y^{(i,r)} - (\mathbf{w}^{(i)})^T \mathbf{x}^{(i,r)} \right)^2$$

- ▶ loss can also be estimated from a reward signal

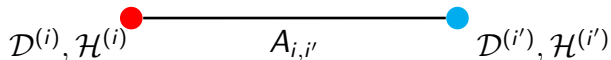
# Edges of an FL network

- ▶ FL network consists of **undirected weighted** edges  $\mathcal{E}$ .
- ▶  $\{i, i'\} \in \mathcal{E}$  signifies a **similarity** between devices  $i$  and  $i'$ .
- ▶ **quantify similarity using edge weight**  $A_{i,i'} > 0$ .
- ▶ notion of similarity depends on FL application .
- ▶ we view edges primarily as a **design choice**.



# Effect of Placing an Edge

FL algorithms are executed over a FL network



placing an edge  $\{i, i'\} \in \mathcal{E}$  has two consequences:

- ▶ there must be communication channel between devices  $i, i'$  (edge weight  $A_{i,i'} \approx$  channel capacity).
- ▶ model parameters at  $i, i'$  are forced to be similar.

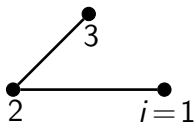
# Connectivity of an FL network

consider an FL network with graph  $\mathcal{G}$ .

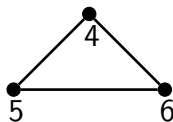
- ▶  $\mathcal{G}$  is **connected** if there is a path between any  $i, i' \in \mathcal{V}$ .
- ▶ a **component**  $\mathcal{C} \subseteq \mathcal{V}$  is a connected subgraph with no edges between  $\mathcal{C}$  and  $\mathcal{V} \setminus \mathcal{C}$ .
- ▶ the **neighborhood** of  $i \in \mathcal{V}$  is  $\mathcal{N}^{(i)} := \{i' \in \mathcal{V} : \{i, i'\} \in \mathcal{E}\}$ .
- ▶ **weighted node degree** of  $i$  is  $d^{(i)} := \sum_{i' \in \mathcal{N}^{(i)}} A_{i,i'}$ .
- ▶ **maximum node degree** is  $d_{\max} := \max_{i \in \mathcal{V}} d^{(i)}$ .

# Connectivity of an FL network- Example

component  $\mathcal{C}^{(1)}$



component  $\mathcal{C}^{(2)}$



- ▶ FL network containing  $n=6$  nodes.
- ▶ uniform edge-weights,  $A_{i,i'} = 1$  for all  $\{i, i'\} \in \mathcal{E}$ .
- ▶ two components  $\mathcal{C}^{(1)} = \{1, 2, 3\}$ ,  $\mathcal{C}^{(2)} = \{4, 5, 6\}$ .
- ▶  $d^{(1)} = 1$ ,  $\mathcal{N}^{(2)} = \{1, 3\}$ ,  $d_{\max} = 2$ .

# From FL network to FL system

each node  $i \in \mathcal{V}$ ,

- ▶ can access local dataset  $\mathcal{D}^{(i)}$ ,
- ▶ maintains model parameters  $\mathbf{w}^{(i)}$
- ▶ sends/receives messages from neighbors  $\mathcal{N}^{(i)}$ .

an FL algorithm specifies *when* and *how* these model parameters are updated.

# FL Algorithms

each node  $i$  uses current model parameters  $\mathbf{w}^{(1,t)}, \dots, \mathbf{w}^{(n,t)}$  to compute new model parameters  $\mathbf{w}^{(i,t+1)}$ ,

$\mathbf{w}^{(i,t+1)} = \mathcal{F}^{(i)}(\mathbf{w}^{(1,t)}, \dots, \mathbf{w}^{(n,t)})$  at time instants  $t = 0, 1, \dots$

the node-wise operator  $\mathcal{F}^{(i)}$  involves

- ▶ local model updates (e.g., via gradient steps)
- ▶ sharing model parameters across edges of FL network.

# What's Next?

L2- “FL Design Principle” introduces generalized total variation minimization (GTVMin) as our main design principle for FL algorithms.

We use GTVMin to guess useful choices for the node-wise update operator  $\mathcal{F}^{(i)}$  that define an FL algorithm.

# References

- ▶ AJ, “Machine Learning: The Basics,” Springer, 2022.
- ▶ AJ, “Federated Learning: From Theory to Practice,” Springer, 2026.
- ▶ AJ et.al., “The Aalto Dictionary of Machine Learning,” github repo, 2026.