Dalle Molle Institute for
Artificial Intelligence
USI – SUPSI

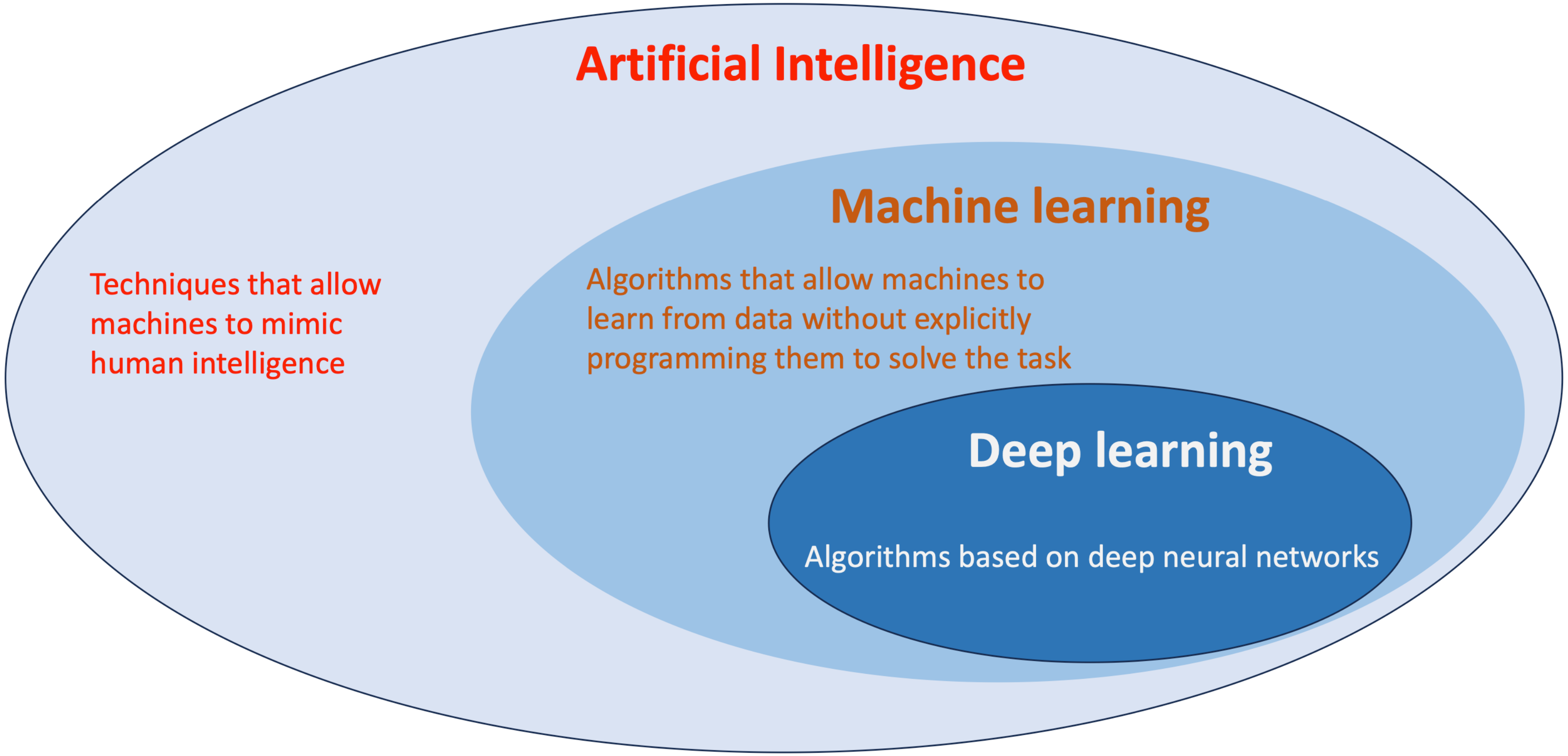
Deep Learning for System Identification

Marco Forgione
SUPSI – Scuola Universitaria Professionale della Svizzera Italiana
IDSIA – Dalle Molle Institute for Artificial Intelligence

marco.forgione@supsi.ch

SUPSI

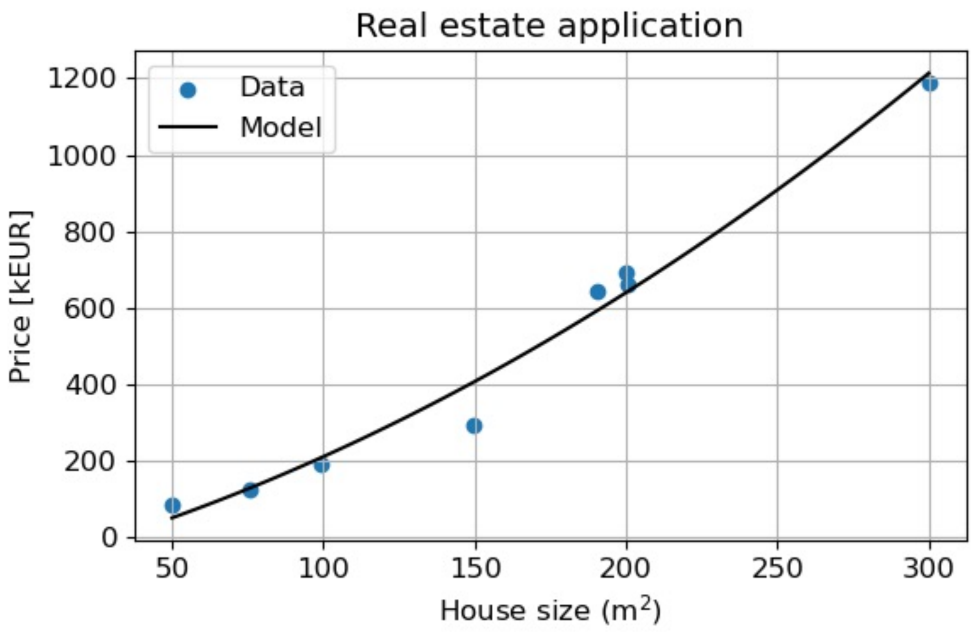
Artificial Intelligence, Machine Learning, Deep Learning



Machine learning: Regression

- Dataset: $D = \{(x_i, y_i)\}_{i=1}^{N}$
- Model structure: $\hat{y} = M(x; \theta)$
- Loss: $\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \|y_i - \hat{y}_i(\theta)\|^2$

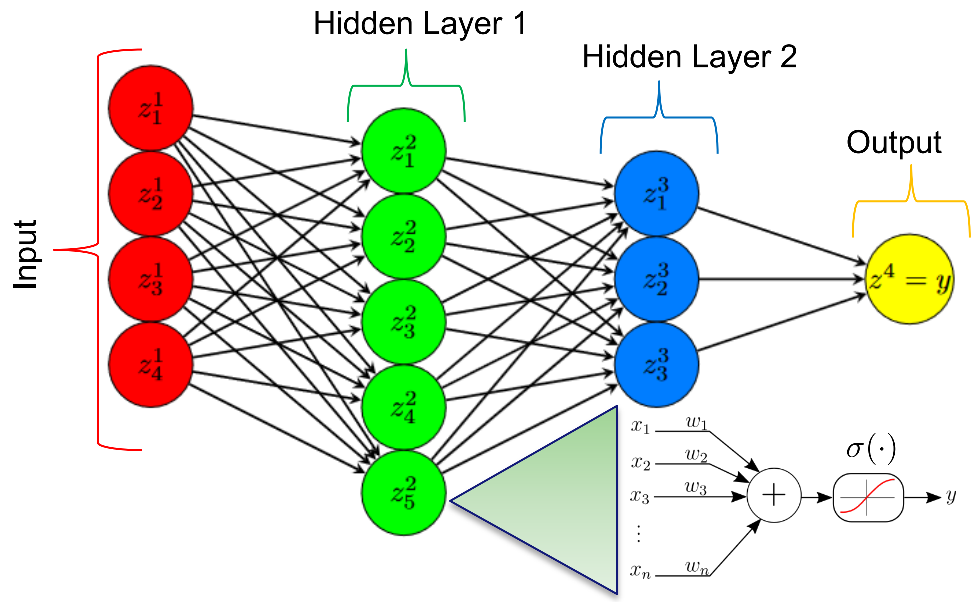
$$\boxed{\hat{\theta} = \arg\min_{\theta} \mathcal{L}(\theta)}$$



- Expressive model structures like neural networks
- Iterative optimization, often gradient-based
- Efficient software for automatic differentiation
- Still, it is just *glorified curve fitting*…

Feed-Forward Neural Networks

Just a particular model structure



$$z_1^2 = \sigma\left(\sum_{j=1}^{4} w_{1,j}^1 z_j^1 + b_1^1\right)$$

$$z_2^3 = \sigma\left(\sum_{j=1}^{5} w_{2,j}^2 z_j^2 + b_2^2\right)$$

$$y = z^4 = \sum_{j=1}^{3} w_{1,j}^3 z_j^3 + b^3$$

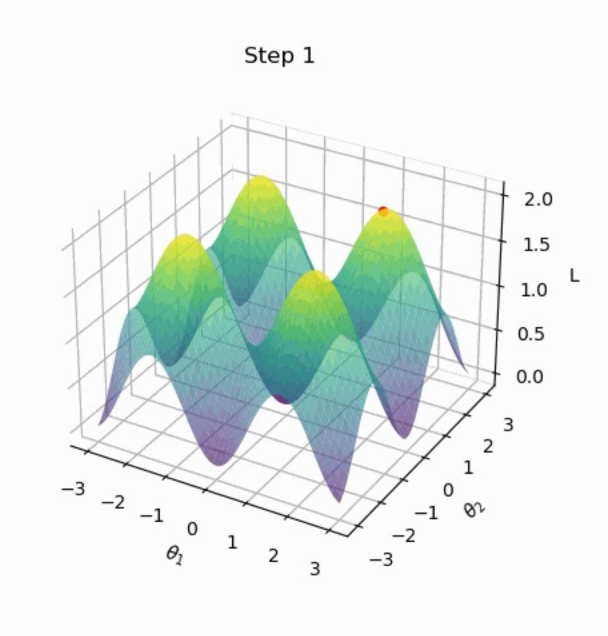
$$y = W_3\sigma\big(W_2\sigma(W_1x + b_1) + b_2\big) + b_3 = \mathrm{FF}(x;\theta)$$

- Linear blocks interleaved by element-wise non-linearities (activation functions)
- Non-linearities essential for expressiveness

Gradient Descent

1. **Initialize Parameters**: $\theta^{(0)}$
2. **for** $k = 0, 1, \ldots$: **until** maximum number of iterations or convergence **do**:
   (a) **Compute Gradient**: $\nabla_\theta \mathcal{L}(\theta^{(k)})$
   (b) **Update Parameters**:

$$\theta^{(k+1)} = \theta^{(k)} - \gamma \cdot \nabla_\theta \mathcal{L}(\theta^{(k)})$$



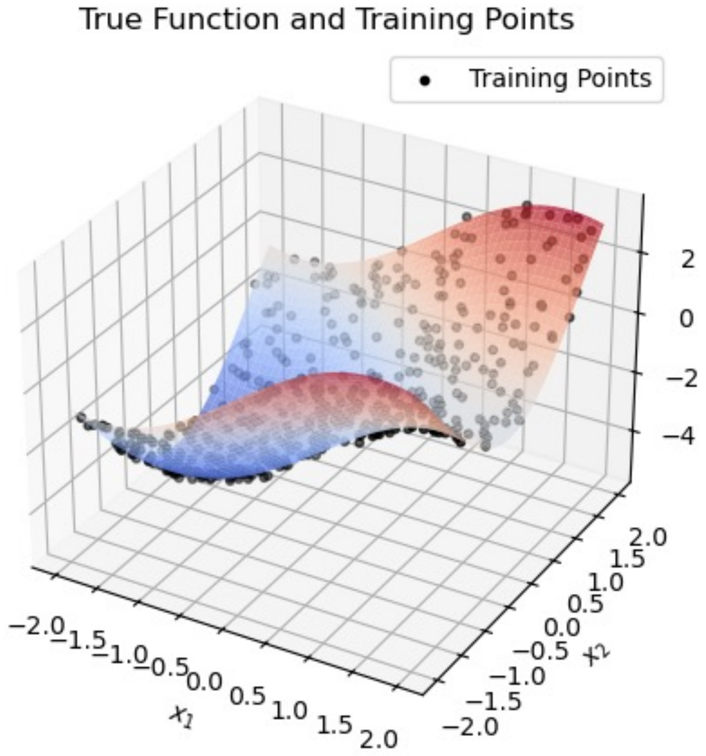
Local convergence to one of the several minima is OK!

- Mini-batching: at each iteration, compute loss & gradients on a subset of the dataset
- Variants of plain gradient descent like Adam are now more common
- Second-order methods like (L)-BFGS sutable for problems up to medium scale
  - most sysid and scientific machine-learning problems

Example: synthetic toy dataset

Consider the 2D function $f : \mathbb{R}^2 \to \mathbb{R}$

$$f(x) = 2\sin(x_1) - 3\cos(x_2)$$
$$x \in [-2, 2]^2 \subset \mathbb{R}^2$$

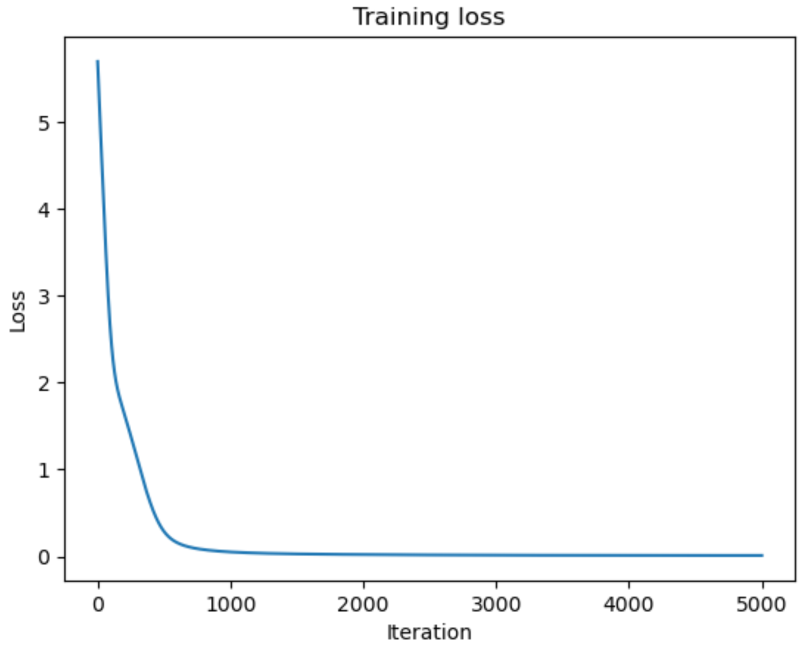


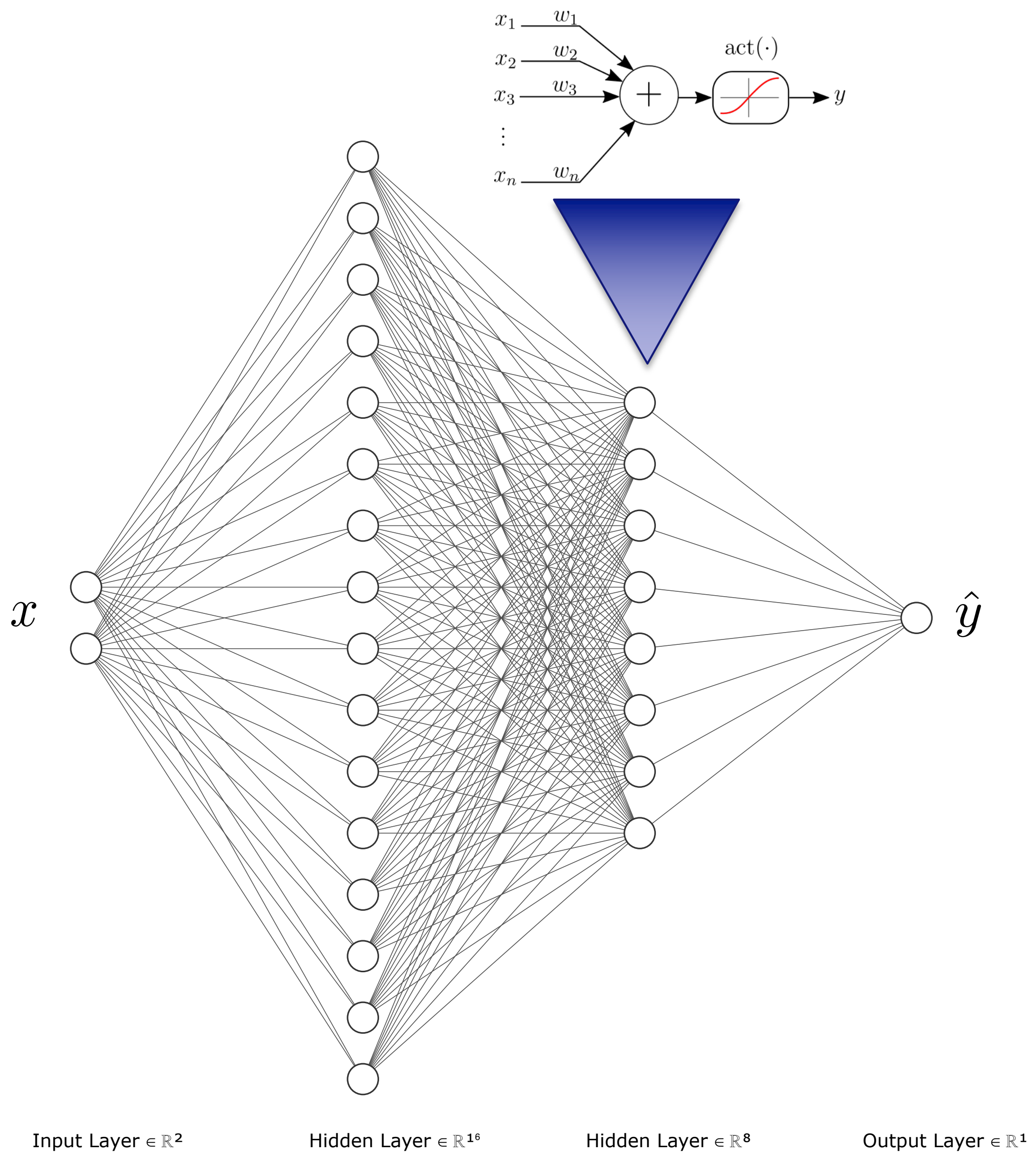
- Training and test datasets: 500 points uniformly sampled in the domain.
- Additive noise with standard deviation 0.1

Feed-forward neural network

- 2 inputs, 1 output - this is the structure of $f(x)$
- 2 hidden layers with [16, 8] neurons
- tanh non-linearities

Training with 5000 iterations of Adam…

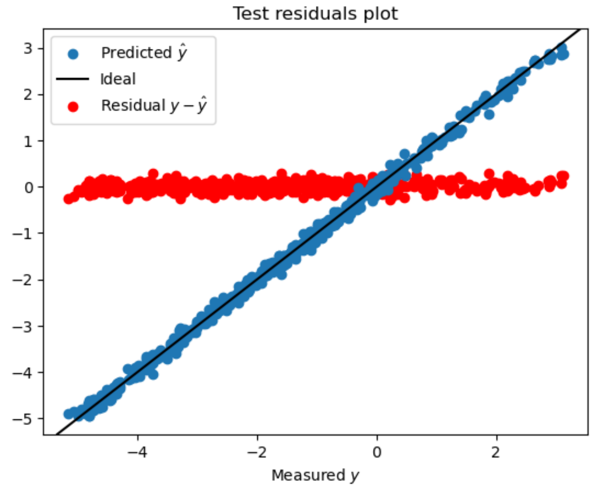


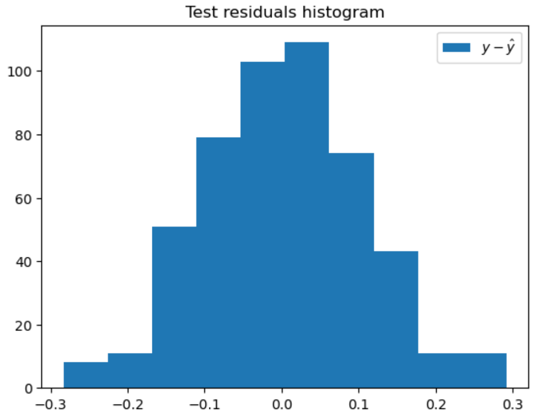


Model evaluation

It is common to inspect on the test dataset:

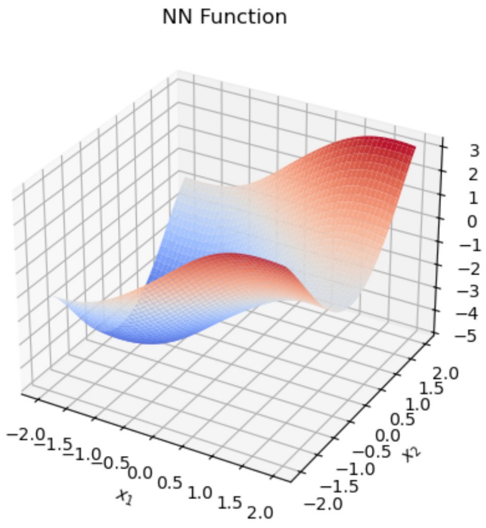
- Predictions and residuals vs measured output (left)
- Histogram of residuals (right)
- Keep on following good statistical practices!

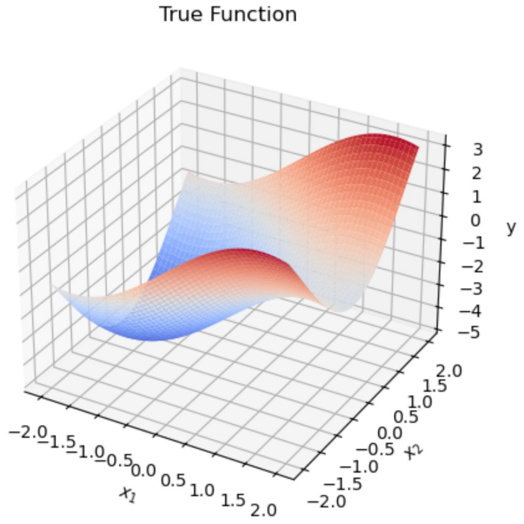




In this 2D toy example, we can also visualize:

- The learned function over a grid (left)
- The known true function (right)





Deep Learning for System Identification

- Feed-forward nets fed by lagged input/outputs are directly applicable for NARX regression

$$\hat{y}(k) = \mathrm{FF}\Big(\overbrace{y(k-1), \ldots, y(k-n_a), u(k), u(k-1), \ldots, u(k-n_b-1)}^{=x(k)}; \theta\Big)$$

- State-space models with neural-network update/output maps:
  - Also known as Recurrent Neural Networks

$$\begin{aligned} x(k+1) &= \mathrm{FF}_x(x(k), u(k); \theta) \\ y(k) &= \mathrm{FF}_y(x(k); \theta) \end{aligned}$$

What we won’t cover

- Advanced architectures
  - physics-informed
  - grounded on system theory (e.g., stable by design)
  - transformers, diffusion, foundation models,...

- Optimization
  - mini-batching (to handle large datasets)
  - second-order, non-smooth, constrained…
  - automatic differentiation details

- Theoretical aspects
  - learning theory
  - uncertainty quantification
  - …

Software for Deep Learning

An essential aspect is the software implementation. Non-exaustive options are:

- PyTorch and JAX: libraries on top of Python
- Julia: a programming language focused on numerical computations
- MATLAB: you should know it already…

We will try out JAX!

Literature

A machine learning reference book

- Murphy, Kevin P. *Probabilistic machine learning: an introduction*. MIT press, 2022.
- Murphy, Kevin P. *Probabilistic machine learning: Advanced topics*. MIT press, 2023

Deep learning in system identification (my biased perspective)

- Forgione, Marco, and Dario Piga. "Continuous-time system identification with neural networks: Model structures and fitting criteria." *European Journal of Control* 59 (2021): 69-81.
- Beintema, Gerben I., Maarten Schoukens, and Roland Tóth. "Deep subspace encoders for nonlinear system identification." *Automatica* 156 (2023): 111210.
- Bemporad, Alberto. "An L-BFGS-B Approach for Linear and Nonlinear System Identification Under ℓ_1 and Group-Lasso Regularization." *IEEE Transactions on Automatic Control* (2025).
- Pillonetto, Gianluigi, Aleksandr Aravkin, Daniel Gedon, Lennart Ljung, Antonio H. Ribeiro, and Thomas B. Schön. "Deep networks for system identification: a survey." *Automatica* 171 (2025): 111907.