2 Fundamental Concepts of Machine Learning: Introduction

Leon Herrmann

Stefan Kollmannsberger

Chair of Data Engineering in Construction

Bauhaus-Universität Weimar

Deep Learning in Computational Mechanics – an introductory course, Herrmann et al. 2025

website

book

Contents

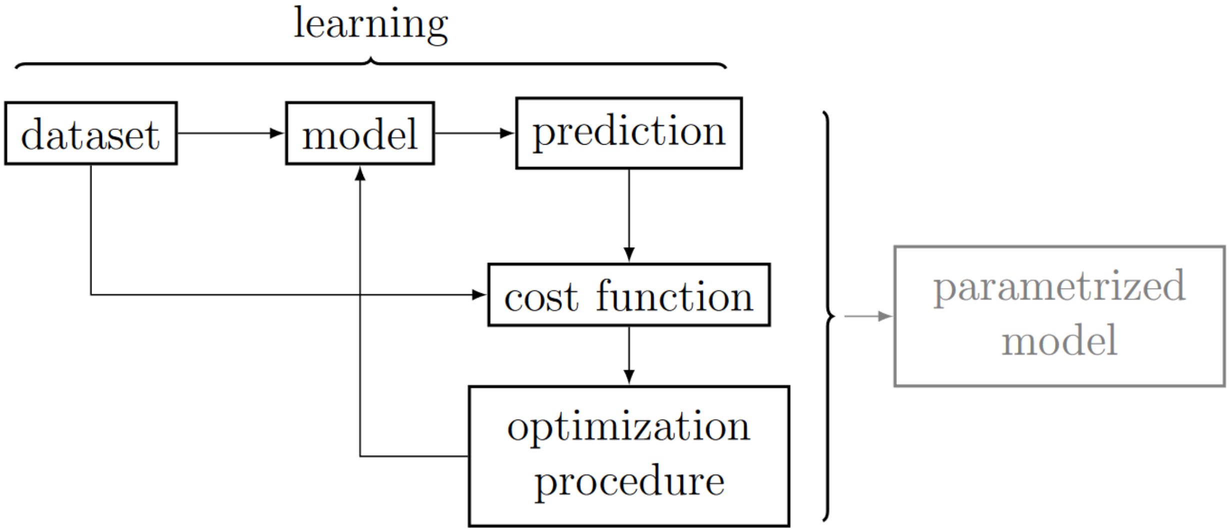
2.1 Definition

”a computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks T, as measured by P, improves with experience E”

Machine Learning, Mitchell 1997

Most machine learning algorithms are composed of

- Dataset
- Parametrized model
- Cost function
- Optimization procedure



2.2 Data Structure

Specific dataset (sometimes called design matrix)

$$\boldsymbol{X} = \begin{array}{c} \\ \text{example } 1 \\ \text{example } 2 \\ \vdots \\ \text{example } m \end{array} \begin{array}{c} \begin{array}{cccc} \text{feature } 1 & \text{feature } 2 & \cdots & \text{feature } n \end{array} \\ \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix} \end{array}$$

Examples

- Can be different images, where its features are its pixel values, $n =$ channels $\times$ pixels
- Can be different houses, where its features are ist properties such as area, number of rooms, age

Notation

- Design matrix $\boldsymbol{X}$
- Design vector of a single example i (1 sample/example) $\boldsymbol{x}_i = [x_{i1}, x_{i2}, \dots, x_{in}]^T$

2.3 Types of Learning

$$\boldsymbol{X} = \begin{array}{c} \\ \text{example } 1 \\ \text{example } 2 \\ \vdots \\ \text{example } m \end{array} \begin{array}{c} \begin{array}{cccc} \text{feature } 1 & \text{feature } 2 & \cdots & \text{feature } n \end{array} \\ \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix} \end{array}$$

Supervised learning

- Algorithm learns from a labeled dataset. Each sample $\boldsymbol{X}_i$ has an accompanying target y_i
- Example: A model learns to distinguish between dogs and cats via annotated images

Unsupervised learning

- Algorithm finds a structure or pattern in the data. This is typically in the form of a probability distribution
- Example: Anomaly detection, i.e., the identification of irregularities in otherwise regular patterns. For example in the detection of tumors in medical imaging

Semi-supervised learning

- Combination of supervised and unsupervised learning, i.e., the data is partly labeled to improve the unsupervised learning.
- Example: The learning of the tumor identification is improved by using some labeled data.

Reinforcement learning

- Interaction between an algorithm and an environment, improving the algorithm to maximize an expected average reward. Common in game-like environments
- Example: The stock market, where more actions with higher rewards are learned

2.4 Machine Learning Tasks

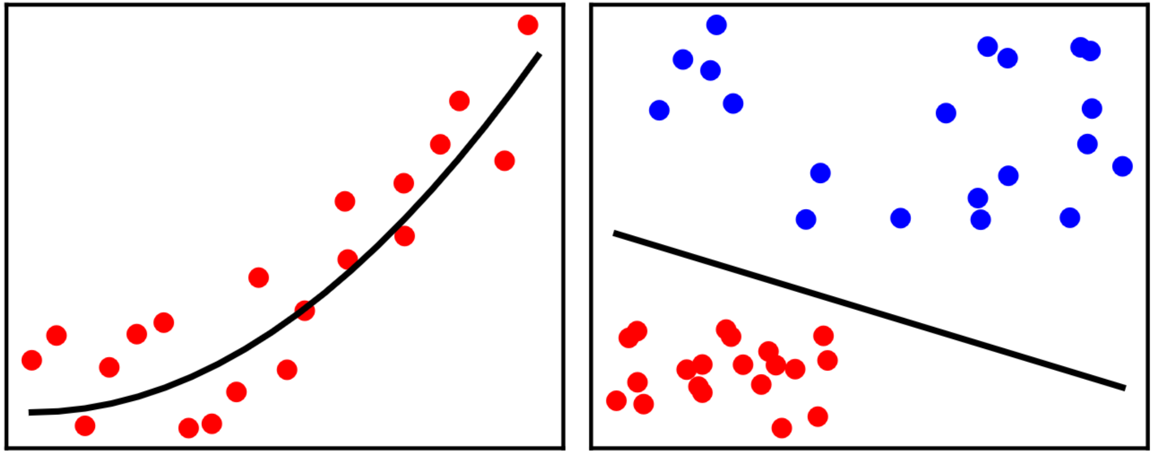
Regression

- Prediction of a numerical value via a (real-valued) mapping between input and output
- Example: Prediction of house prices from criteria like area, number of rooms, age

Classification

- Prediction of a discrete category via a mapping between input and a (discrete) category
- Example: Classification of images in cats and dogs

Classification can be regarded as discrete regression.



Regression   Classification

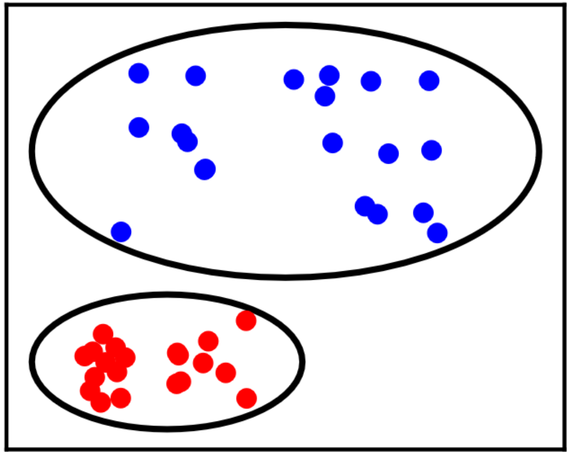
2.4 Machine Learning Tasks

Clustering

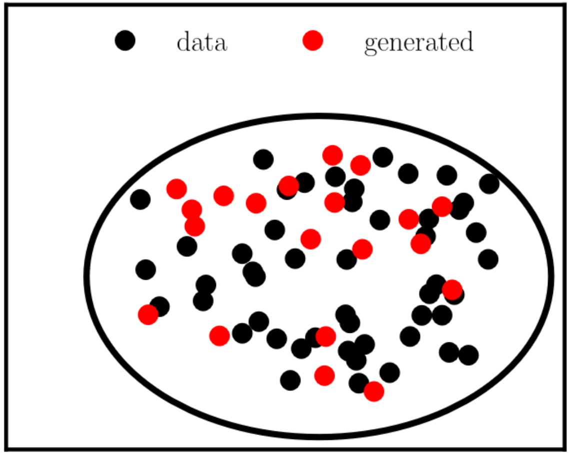
- Discovers similarities between data and creates discrete clusters (unsupervised)
- Example: Identification of similar customer groups

Generative modeling

- Generate new data points that resemble a given dataset (without simply reproducing given data points)
- Example: Generate new rim designs given a set of rims



Clustering



Generative modeling

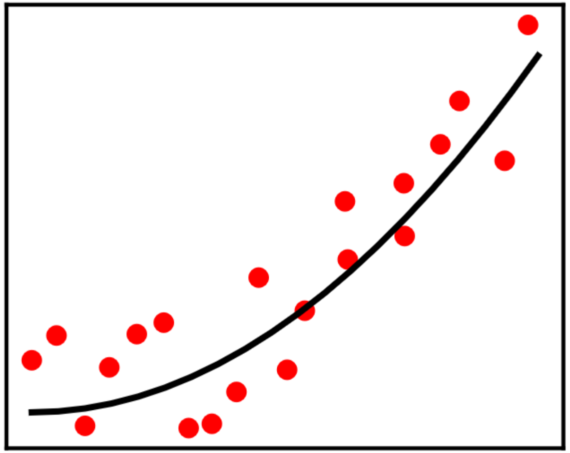
Machine Learning Algorithms

Machine Learning in Additive Manufacturing: State-of-the-Art and Perspectives, Wang et al. 2020

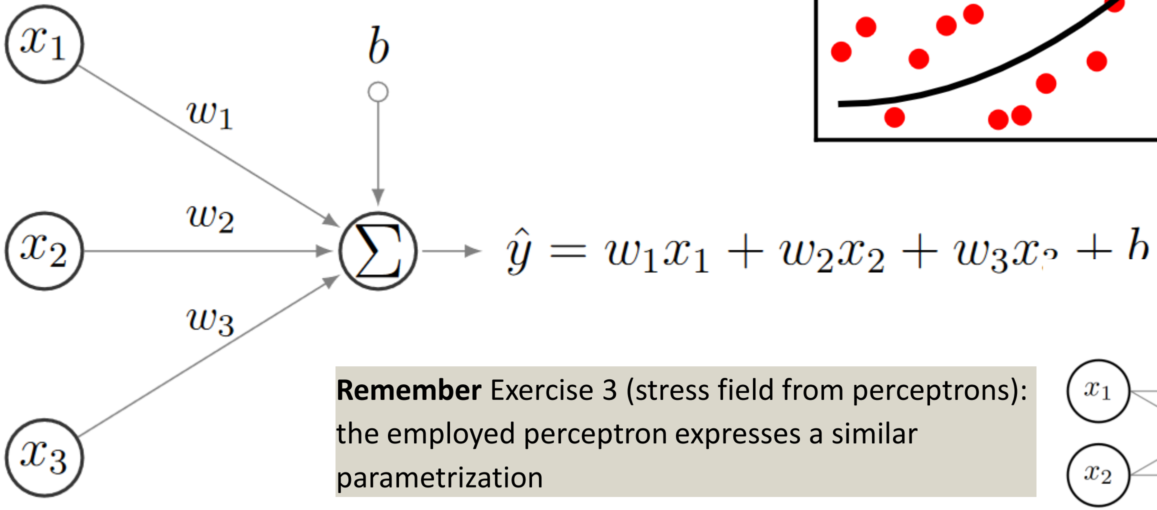
	Classifications	Algorithms	Tasks
	Supervised	Decision trees	Classification
		Random forest	Classficiation, regression
covered in Chapter 7		Support vector machines	Classification, regression
		K-nearest neighbours	Classification
		Bayesian network	Classification
		Gaussian process	Regression
		Multi-gene genetic programming	Regression
		Hidden semi-Markov model	Classification
		Multi-layer perceptron	Classification, regression
covered in Chapter 3		Convolutional neural network	Classification
		Recurrent neural network	Time series prediction (regression)
		Adaptive network-based fuzzy inference system	Regression
covered in Chapter 8		Transformers	Regression, classification, generative modeling
	Unsupervised	Self-organizing map	Clustering
		Deep belief network	Classification
covered in Chapter 7		K-means clustering	Clustering
		Reduced order modeling (POD)	Dimensionality reduction
		Autoencoder	Generative modeling, dimensionality reduction
covered in Chapter 8		Generative adversarial networks	Generative modeling, (classification)
		Diffusion model	Generative modeling
	Semi-supervised	Gaussian mixture model	Clustering

2.5 Linear Regression – Prediction

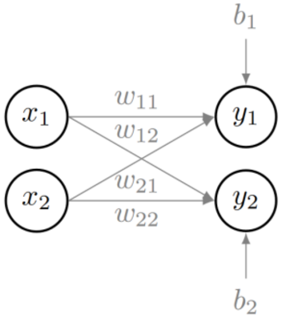
$$\hat{y} = \boldsymbol{w} \cdot \boldsymbol{x} + b = \sum_{j=1}^{n} w_j x_j + b$$



- Target: $\hat{y}$
- Ground truth: $\tilde{y}$ (associated with $\tilde{\boldsymbol{x}}$)
- Example vector: $\boldsymbol{x}$
- Weight vector: $\boldsymbol{w}$
- Bias: b



Remember Exercise 3 (stress field from perceptrons): the employed perceptron expresses a similar parametrization



2.5 Linear Regression – Performance Measurement

Prediction in nd for sample i:

$$\hat{y}_i = \boldsymbol{w} \cdot \boldsymbol{x}_i + b = \sum_{j=1}^{n} w_j x_{ij} + b$$

i is an example/sample and j is a feature

- each feature has an associated weight, which is shared across all samples

Remember the design matrix

$$\boldsymbol{X} = \begin{array}{c|cccc} & \text{feature } 1 & \text{feature } 2 & \cdots & \text{feature } n \\ \hline \begin{array}{c} \text{example } 1 \\ \text{example } 2 \\ \vdots \\ \text{example } m \end{array} & \multicolumn{4}{c}{\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix}} \end{array}$$

Where each example vector is defined as $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \dots, x_{in})^T$

2.5 Linear Regression – Performance Measurement

Prediction in semi-vector notation for sample i

$$\hat{y}_i = \boldsymbol{w} \cdot \tilde{\boldsymbol{x}}_i + b$$

Squared error for a single sample i (with **ground truth** $\tilde{y}_i$)

$$(\tilde{y}_i - \hat{y}_i)^2$$

Mean squared error (MSE) for a dataset X with m samples (including the design vectors x_i of each sample i)

$$\mathcal{L} = \frac{1}{m}\sum_{i=1}^{m}(\tilde{y}_i - \hat{y}_i)^2 = \frac{1}{m}\sum_{i=1}^{m}(\tilde{y}_i - (\boldsymbol{w} \cdot \tilde{\boldsymbol{x}}_i + b))^2$$

Cost function

$$C(\boldsymbol{w}, b) = \mathcal{L} + \cdots$$

Optimization problem

$$\min_{\boldsymbol{w},b} C(\boldsymbol{w}, b) = \min_{\boldsymbol{w},b} \frac{1}{m}\sum_{i=1}^{m}\left(\tilde{y}_i - (\boldsymbol{w} \cdot \tilde{\boldsymbol{x}}_i + b)\right)^2$$

In machine learning optimization is referred to as **learning** when the model is applied to previously unseen problems (i.e., datapoints). This stands in contrast to structural optimization in which one specific design is obtained through optimization.

2.5 Linear Regression – Data Split

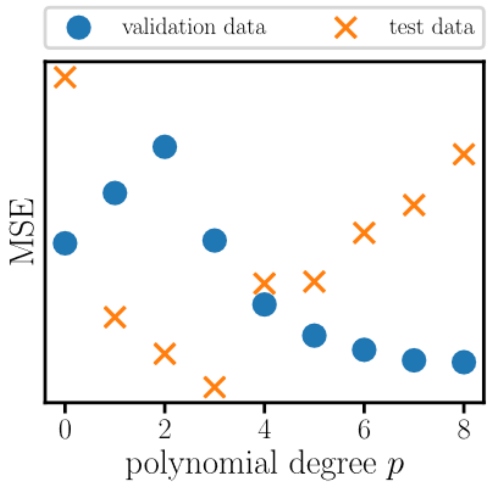
Data is split into

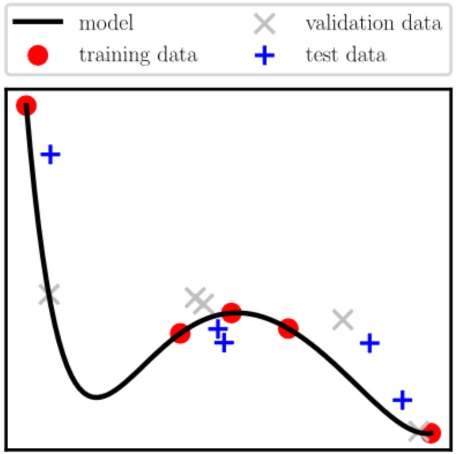
- **Training set** ($\sim 80\%$)
  - to train the model (i.e., find the correct weights and bias)
- **Validation set** ($\sim 10\%$)
  - to validate the training (i.e., evaluate if training is successful, e.g., to detect/avoid **overfitting**)
  - to find the correct (machine learning algorithm) **hyperparameters**
- **Testing set** ($\sim 10\%$)
  - to test/assess the validity of the final model (i.e., weights, bias, and hyperparameters)
  - At this point nothing is allowed to be changed (otherwise testing set becomes validation set)
  - In AI double-blinded challenges are common (test set is released only after handing model to jury)

2.5 Linear Regression – Data Split

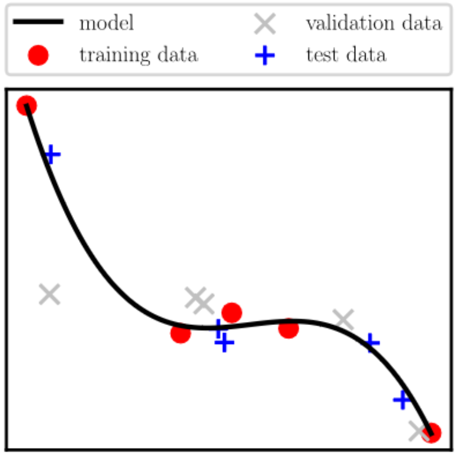
Example

- Fitting of datapoints with a polynomial where the hyperparameter is the polynomial degree p





$p = 8$



$p = 3$

- The validation set shows that $p = 8$ leads to the lowest validation error for the trained model
- The test set shows that some overfitting happened during hyperparameter tuning (this information is not available during/for model development)
- The best model would rely on $p = 3$

2.5 Linear Regression – Optimization

$$\min_{\boldsymbol{w},b} C(\boldsymbol{w},b) = \min_{\boldsymbol{w},b} \frac{1}{m}\sum_{i=1}^{m}\left(\tilde{y}_i - (\boldsymbol{w}\cdot\tilde{\boldsymbol{x}}_i + b)\right)^2$$

Note that $\boldsymbol{X}$ requires a column of ones for the bias b

For a more concise notation let us denote all learnable parameters in a vector $\boldsymbol{\Theta} = (\boldsymbol{w}, b)^T$

This allows to write the model function $\hat{y}_i = \boldsymbol{w}^T\tilde{\boldsymbol{x}}_i + b$ as $\hat{\boldsymbol{y}} = \boldsymbol{X\Theta}$ yielding the minimization

All predictions $\hat{y}_i$ are collected in the vector $\hat{\boldsymbol{y}}$.

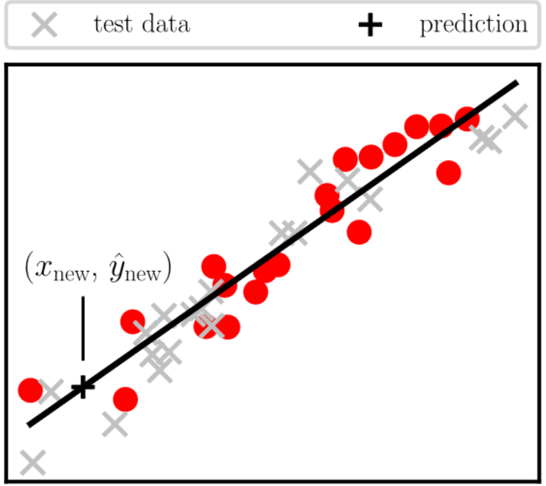
$$\min_{\boldsymbol{\Theta}} C(\boldsymbol{\Theta}) = \min_{\boldsymbol{\Theta}}(\tilde{\boldsymbol{y}} - \boldsymbol{X\Theta})(\tilde{\boldsymbol{y}} - \boldsymbol{X\Theta}) = \min_{\boldsymbol{\Theta}}(\tilde{\boldsymbol{y}}^T\tilde{\boldsymbol{y}} - 2\tilde{\boldsymbol{y}}^T\boldsymbol{X\Theta} + (\boldsymbol{X\Theta})^T\boldsymbol{X\Theta})$$

The minimization is solved by setting the first derivative of C with respect to $\boldsymbol{\Theta}$ to zero (using $\boldsymbol{r} = \tilde{\boldsymbol{y}} - \boldsymbol{X\Theta}$)

$$\frac{1}{2}\frac{\partial \boldsymbol{r}(\boldsymbol{\Theta})^2}{\partial\boldsymbol{\Theta}} = \frac{1}{2}(-2\boldsymbol{X}^T\tilde{\boldsymbol{y}} + 2\boldsymbol{X}^T\boldsymbol{X\Theta}) = -\boldsymbol{X}^T\tilde{\boldsymbol{y}} + \boldsymbol{X}^T\boldsymbol{X\Theta} = 0$$

$$\boldsymbol{X}^T\boldsymbol{X\Theta} = \boldsymbol{X}^T\tilde{\boldsymbol{y}}$$

$$\boldsymbol{\Theta} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\tilde{\boldsymbol{y}}$$



Such a closed form solution is only possible if $\hat{\boldsymbol{y}}$ (or rather $\partial C/\partial\boldsymbol{\Theta}$) is **linear** with respect to $\boldsymbol{\Theta}$

2.5 Linear Regression – Example

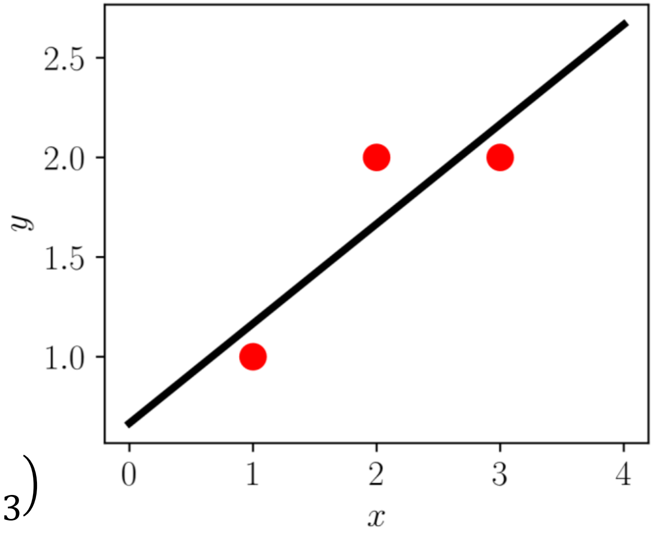
$$\widehat{\boldsymbol{y}} = \mathbf{X\Theta}$$
$$\boldsymbol{\Theta} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\widetilde{\boldsymbol{y}}$$

Given the following values, solve the linear regression

$$\boldsymbol{x} = (1,2,3), \widetilde{\boldsymbol{y}} = (1,2,2)$$

Example matrix (including column of ones for bias)

$$\boldsymbol{X} = \begin{pmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \end{pmatrix}, \boldsymbol{X}^T = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 1 & 1 \end{pmatrix}$$
$$\boldsymbol{X}^T\boldsymbol{X} = \begin{pmatrix} 14 & 6 \\ 6 & 3 \end{pmatrix}, (\boldsymbol{X}^T\boldsymbol{X})^{-1} = \begin{pmatrix} 0.5 & -1 \\ -1 & 2.333 \end{pmatrix}$$



Optimal weight and bias

$$\boldsymbol{\Theta} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\widetilde{\boldsymbol{y}} = (0.5, 0.667)^T$$
$$w = 0.5, b = \frac{2}{3}$$

Contents

2 Fundamental Concepts of Machine Learning: Introduction

Leon Herrmann

Stefan Kollmannsberger

Chair of Data Engineering in Construction

Bauhaus-Universität Weimar

Deep Learning in Computational Mechanics – an introductory course, Herrmann et al. 2025

website

book