Applied Numerical Optimization

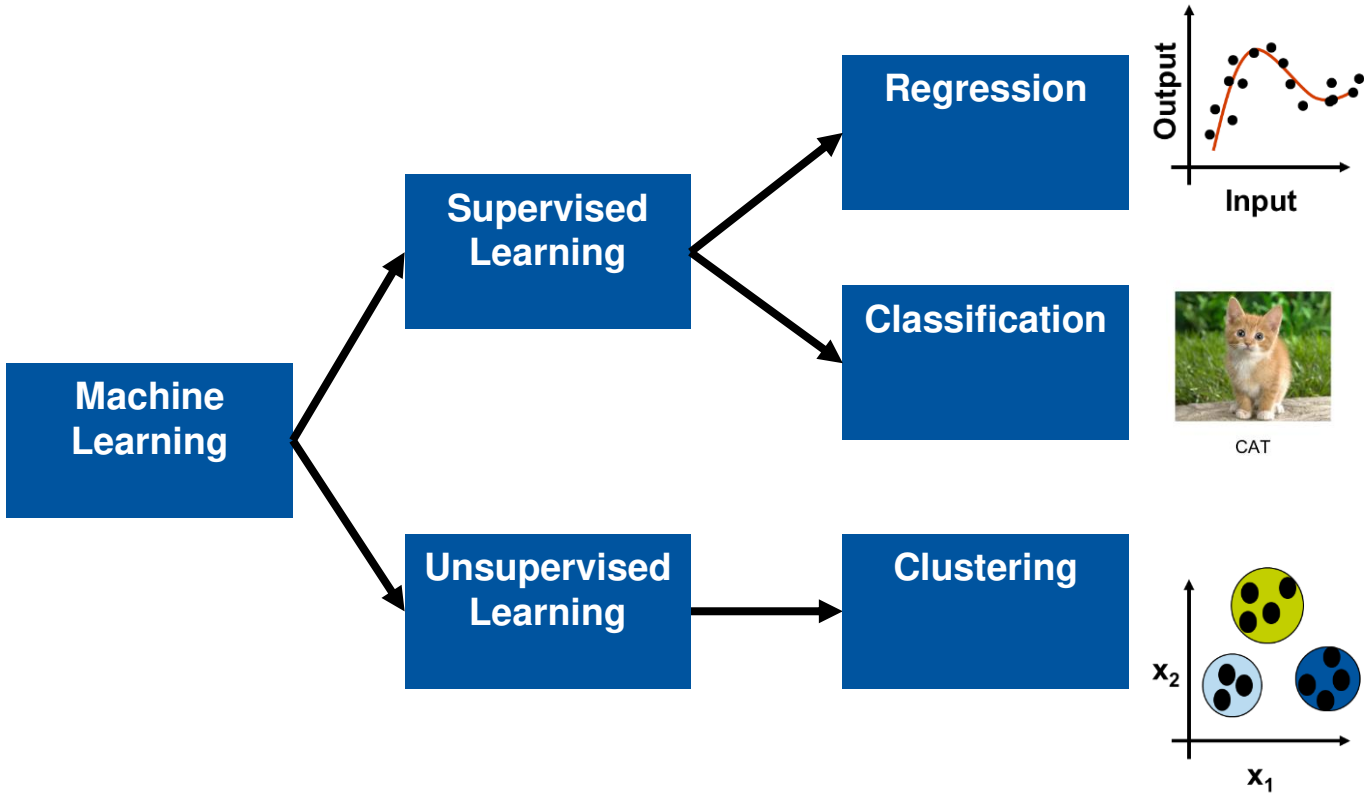
Prof. Alexander Mitsos, Ph.D.

Machine learning and hybrid modeling

Aachener Verfahrenstechnik | RWTH AACHEN UNIVERSITY

Machine Learning[1]

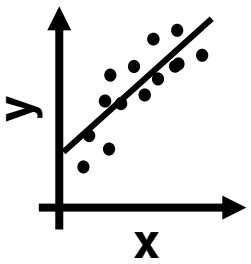
- Programming computers using example data or past experience
- Learning general models from example data
- Key ingredient: **data**
  - formats vary widely
  - labeled?
  - structured?
  - relevant?



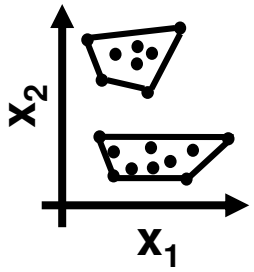
[1] Bishop, C. M. (2009): Pattern Recognition and Machine Learning.

Data Driven Model Structures

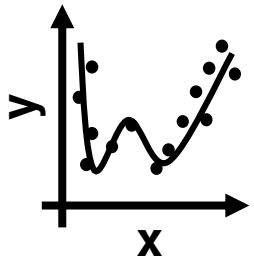
Linear



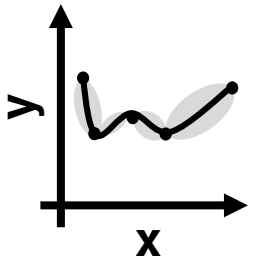
Convex region linear surrogate



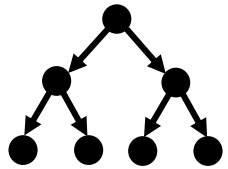
Basis functions



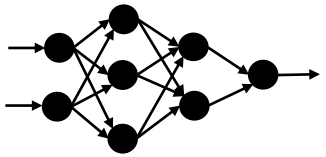
Gaussian processes



Gradient-boosted trees



Artificial neural networks

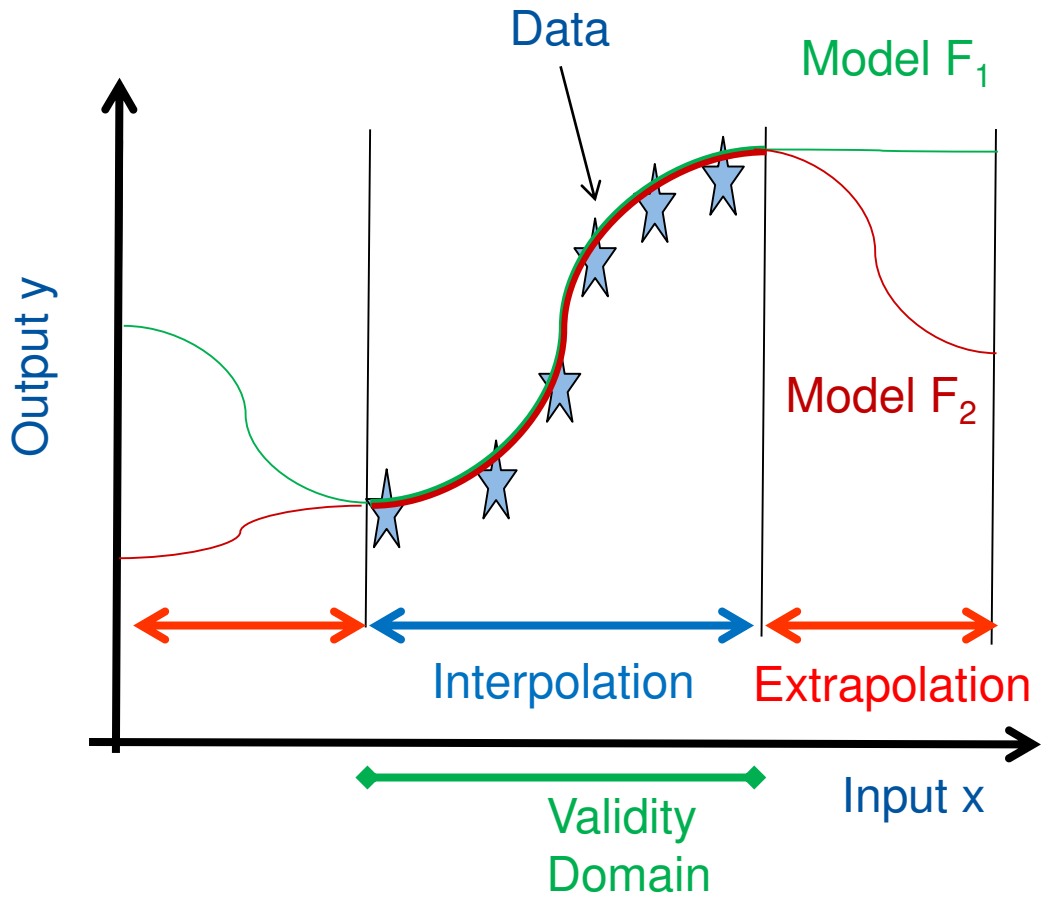


Surrogate model complexity

Interpolation and Extrapolation

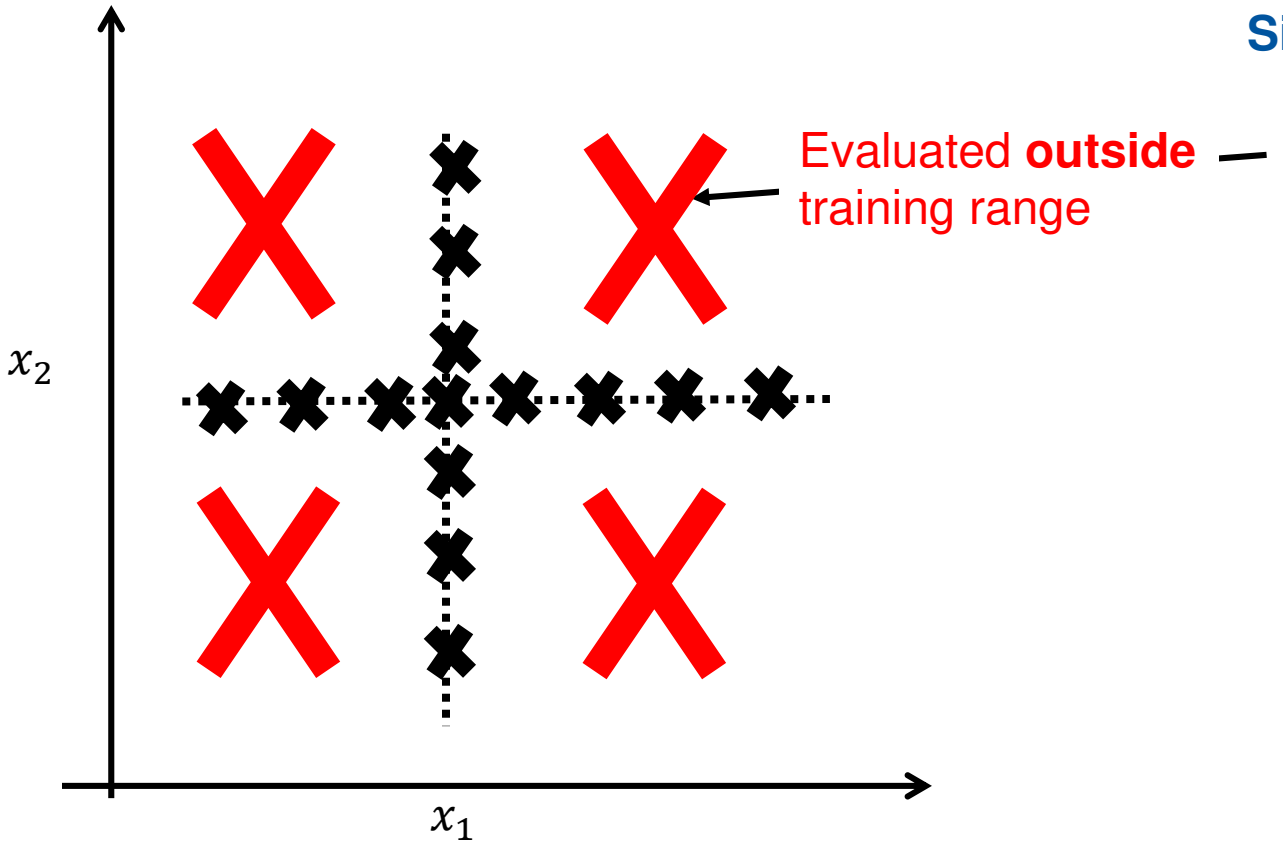
- **Interpolation:** prediction within the convex hull of training data
- **Extrapolation:** prediction outside the convex hull of training data

→Data-driven models cannot extrapolate

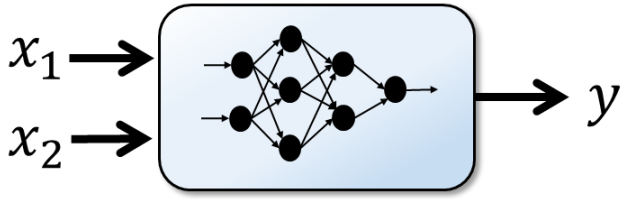


→ Staying in validity domain is essential for black-box models

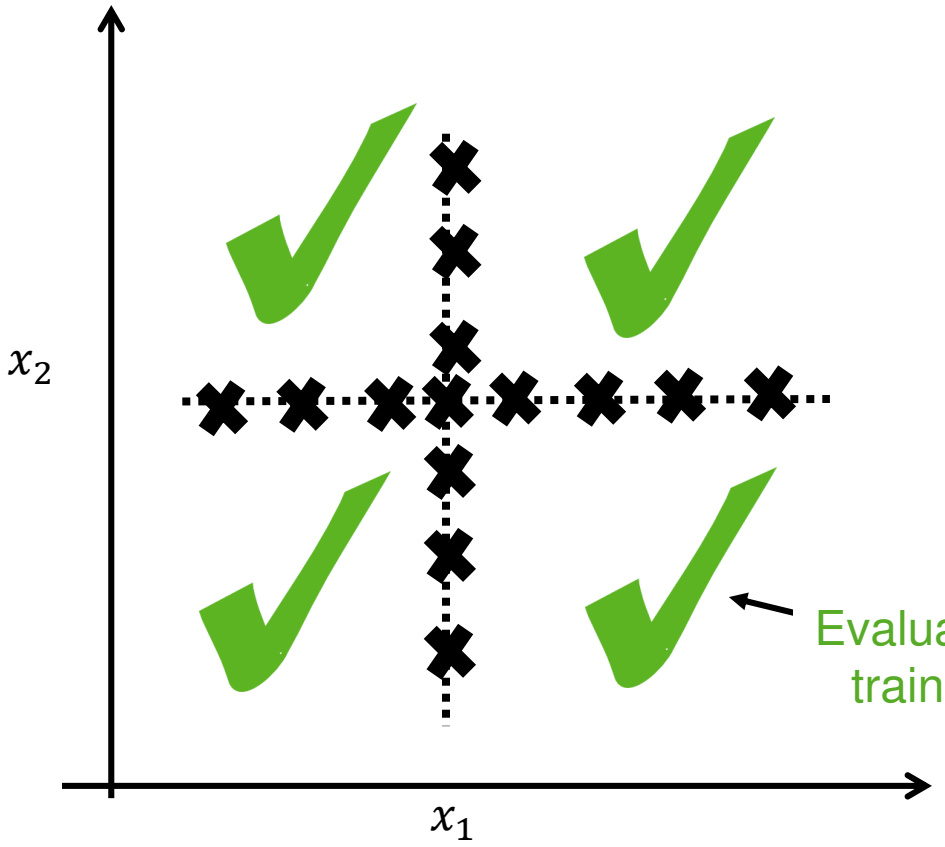
Hybrid Modelling – Combining Mechanistic and Data-driven Models



Single data-driven model

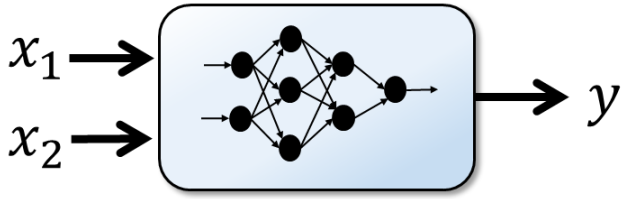


Hybrid modeling – Combining Mechanistic and Data-Driven Models

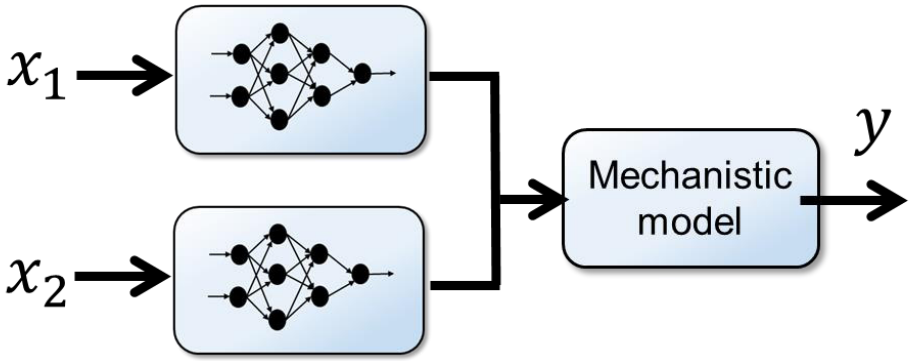


Evaluated **inside** training range

Single data-driven model



Hybrid model structure

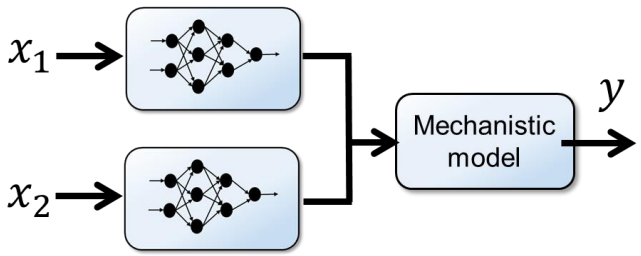


Training/Fitting of Hybrid Models

- Different training approaches possible, e.g.,
  - train hybrid model
  - use different data for each data-driven part
  - train separately mechanistic and data-driven model

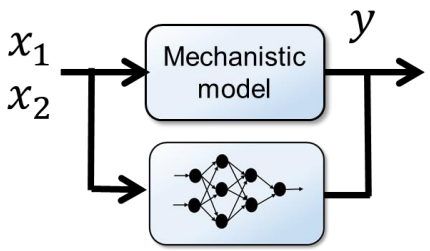
- Different topologies possible, e.g.,:

serial and structured approach



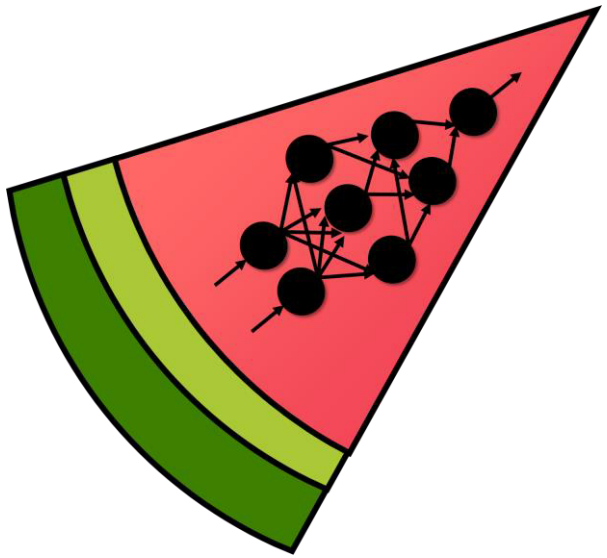
models relations within the mechanistic model

parallel approach



compensates errors of mechanistic model

MeLOn – Machine Learning Models for Optimization



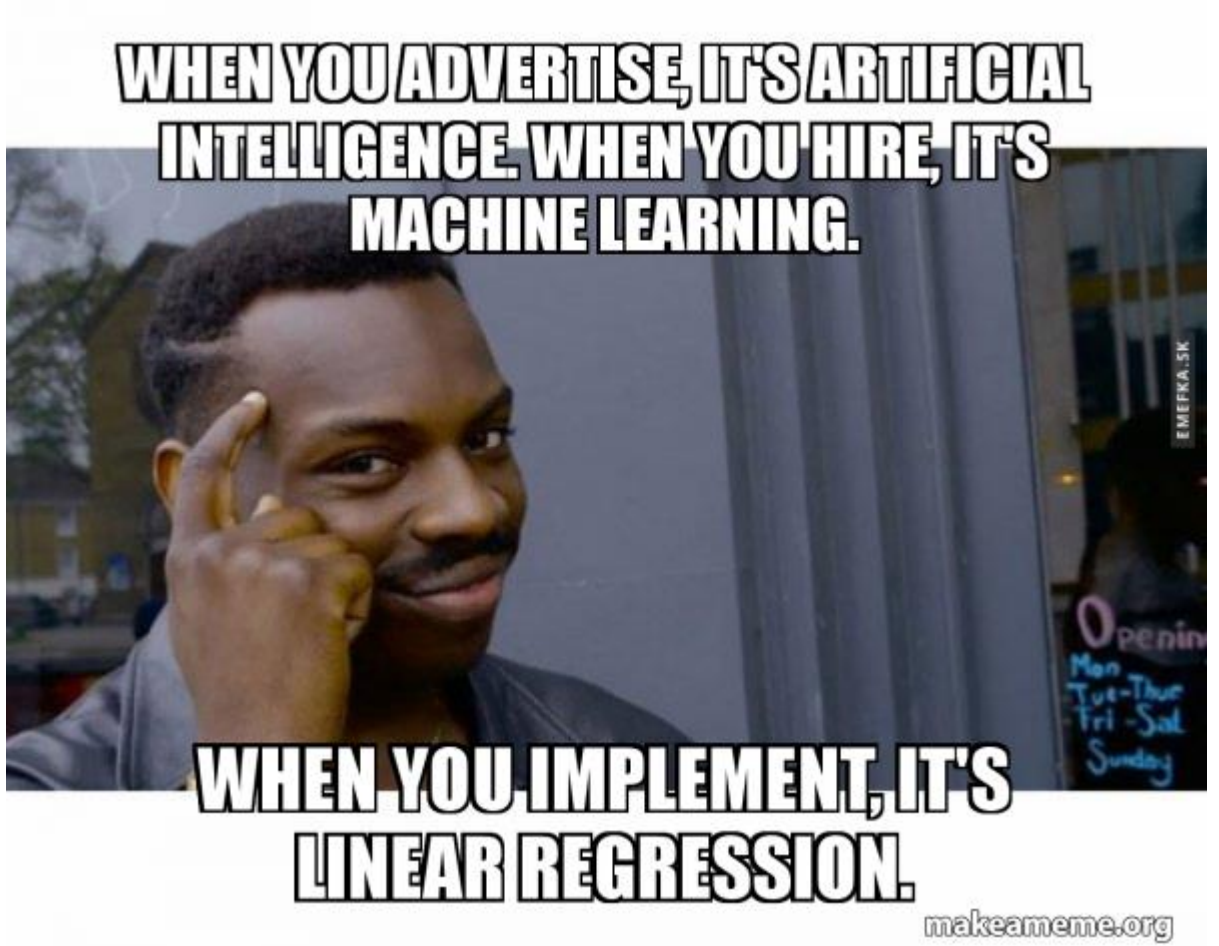
- MeLOn provides machine learning models with training scripts and examples
- The models can be used within our open-source global solver MAiNGO or exported to GAMS
- Interfaces to Keras, TensorFlow, scikit-learn, MATLAB, python
- Includes artificial neural networks, Gaussian processes,…

MeLOn

Open-source
https://git.rwth-aachen.de/avt.svt/public/MeLOn

Check Yourself

- What is machine learning?
- Can data-driven model extrapolate from their training data set?
- Why is it beneficial to use hybrid models?



https://media.makeameme.org/created/when-you-advertise-f81897f53a.jpg

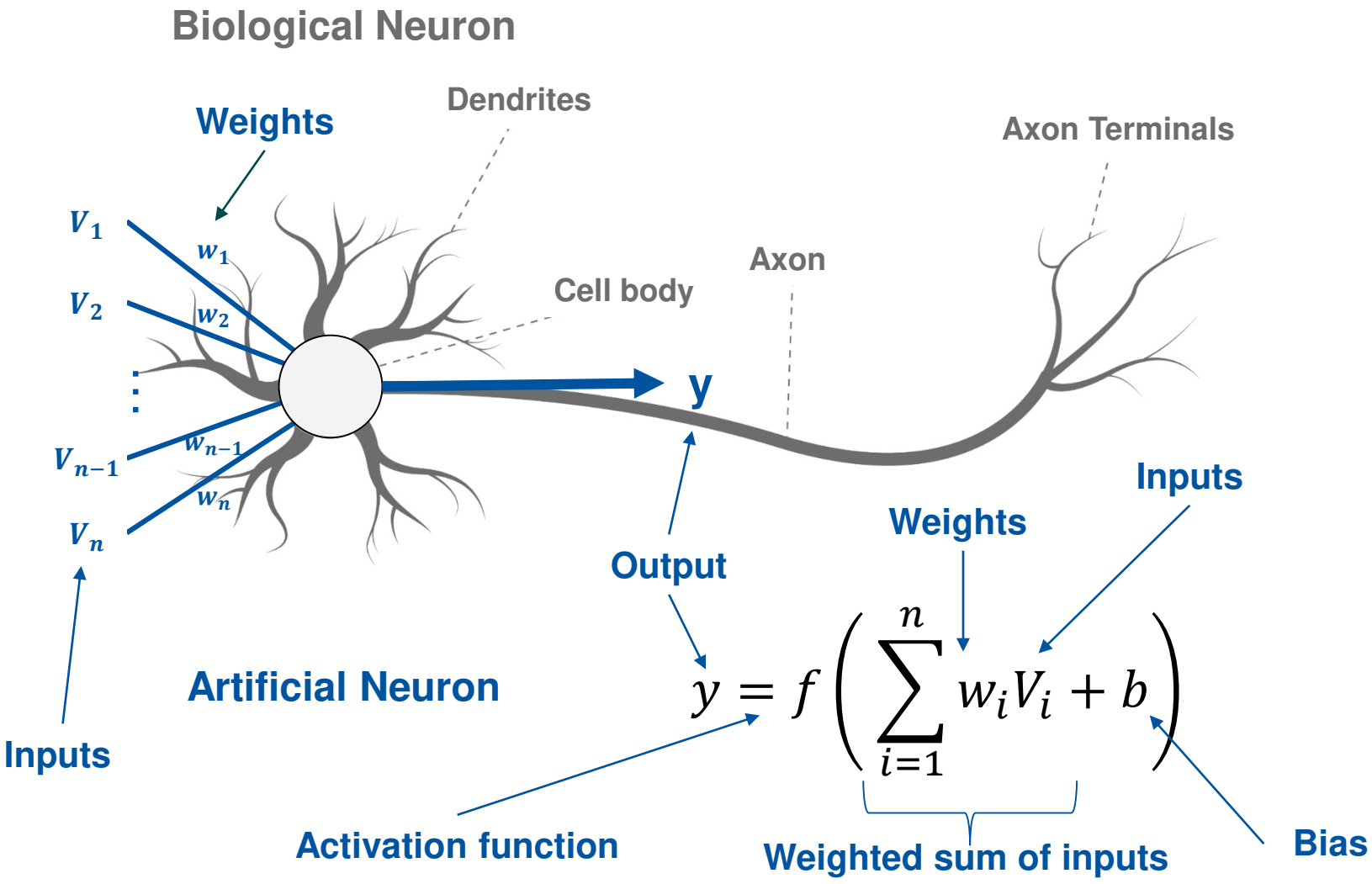
Applied Numerical Optimization

Prof. Alexander Mitsos, Ph.D.

Artificial Neural Networks

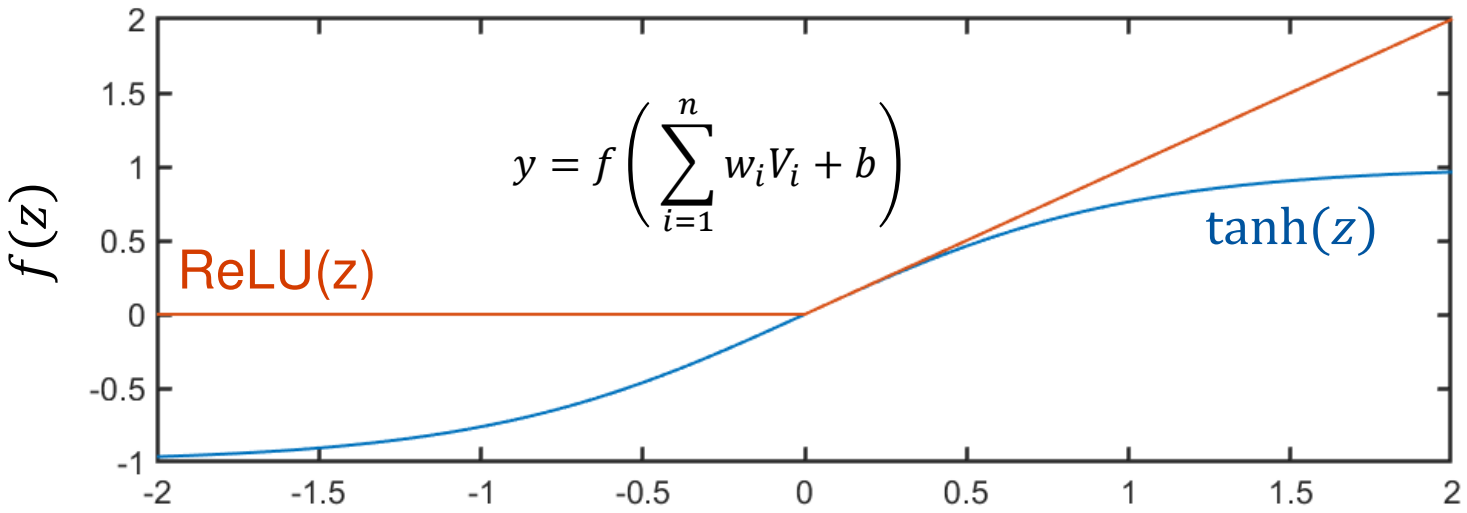
Aachener Verfahrenstechnik | RWTH AACHEN UNIVERSITY

Biological and Artificial Neuron



https://machineslearn.co.uk/wp-content/uploads/2018/01/biological-neuron-1024x438.png

Artificial Neural Networks: Activation Functions

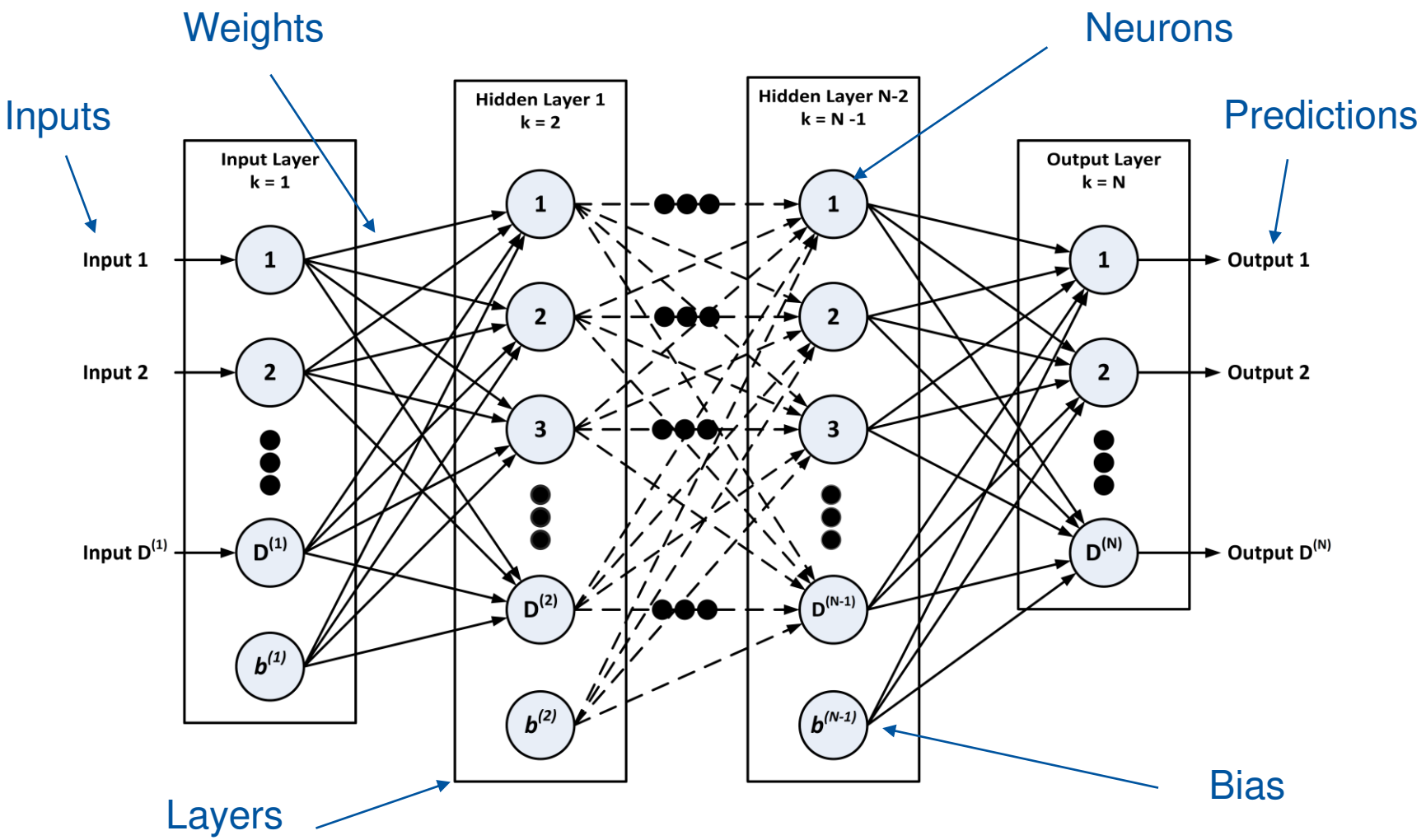


$$z = \sum_{i=1}^{n} w_i V_i + b$$

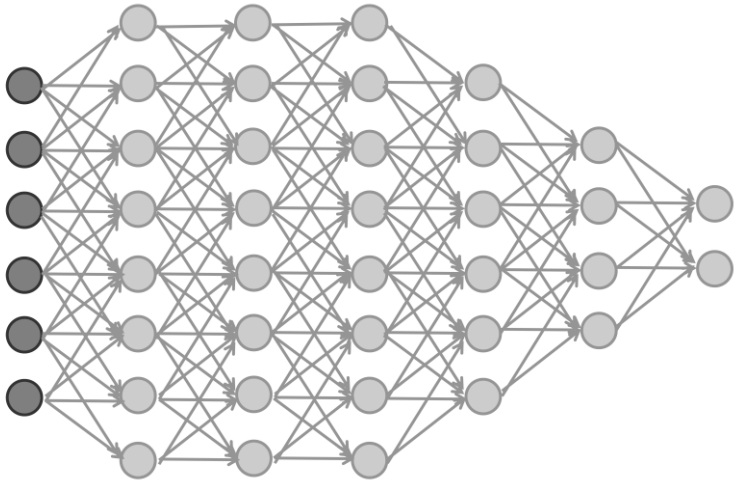
Several alternative activation functions, e.g.,

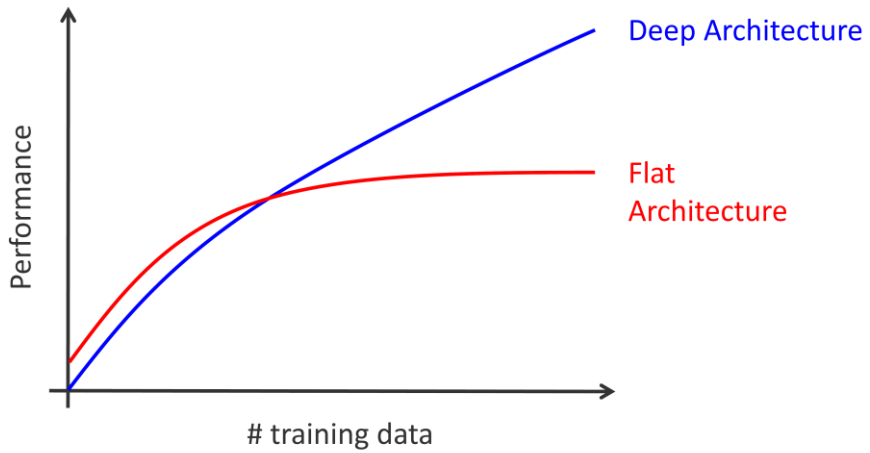
- Hyperbolic tangent function $f(z) = \tanh(z)$
  - Saturates quickly
- Rectified Linear Unit (ReLU) $f(z) = \max(0, z)$
  - Used in deep ANNs

Feed-Forward ANNs



Deep Learning





- Deep networks show better performance for large data sets, e.g., image recognition
  - AlexNet (2012): 8 (learning) layers with 60 mio parameters
  - Inception-v3 (2015): 47 (learning) layers with 25 mio parameters
- Success factors: e.g., GPU computing, ReLU activation function, dropout regularization

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." Nature 521.7553 (2015): 436.

B. W. Schuller. “Deep Learning for Signal Analysis”, Deep Learning Summer School 2018.

Check Yourself

- What are ANNs?
- What is deep learning?

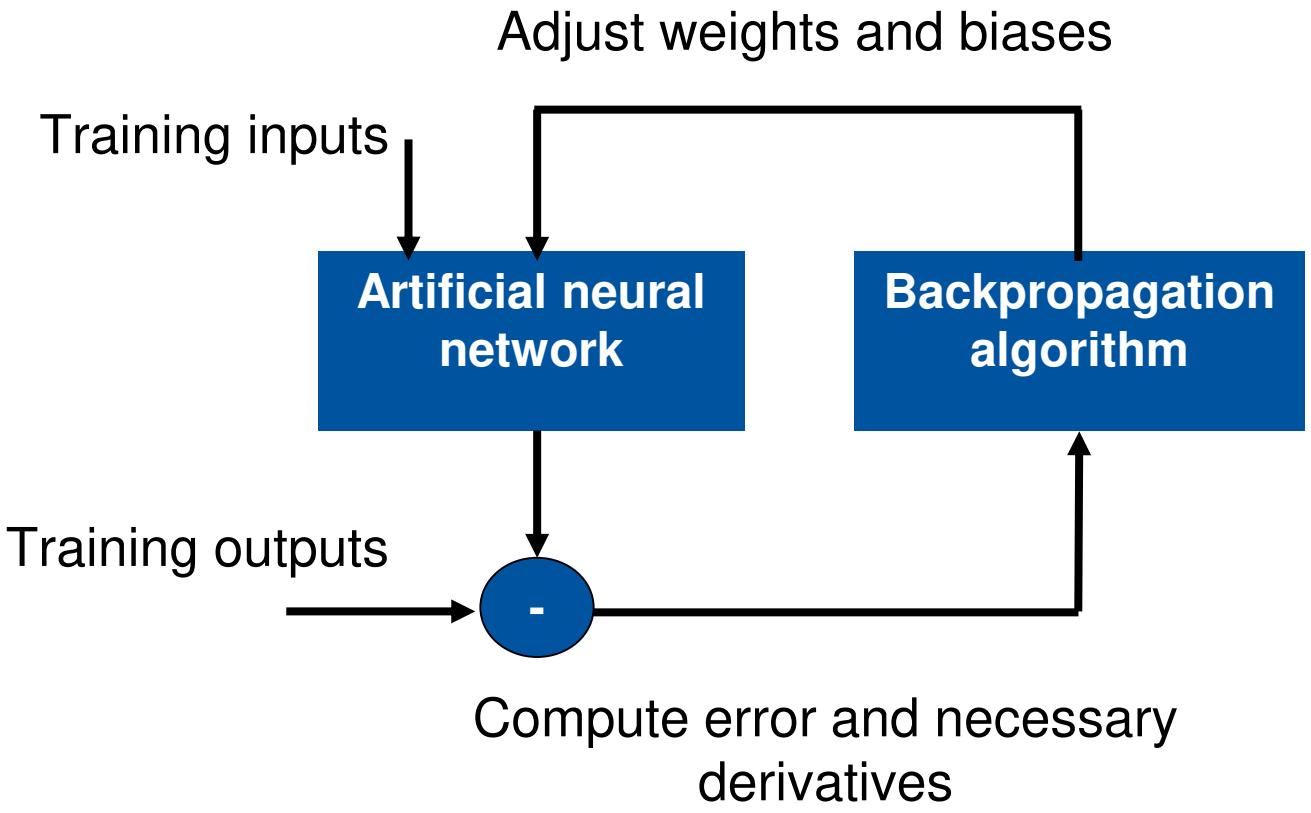
Applied Numerical Optimization

Prof. Alexander Mitsos, Ph.D.

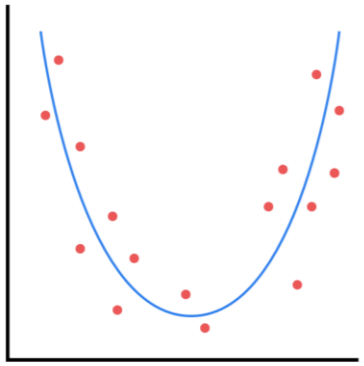
Training of data-driven models: ANN

Aachener Verfahrenstechnik | RWTH AACHEN UNIVERSITY

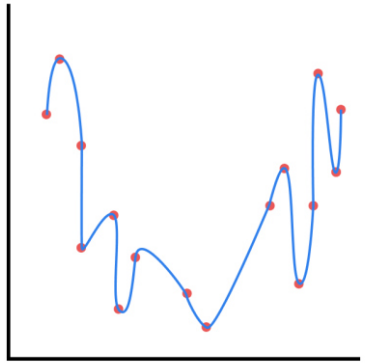
Training of ANNs



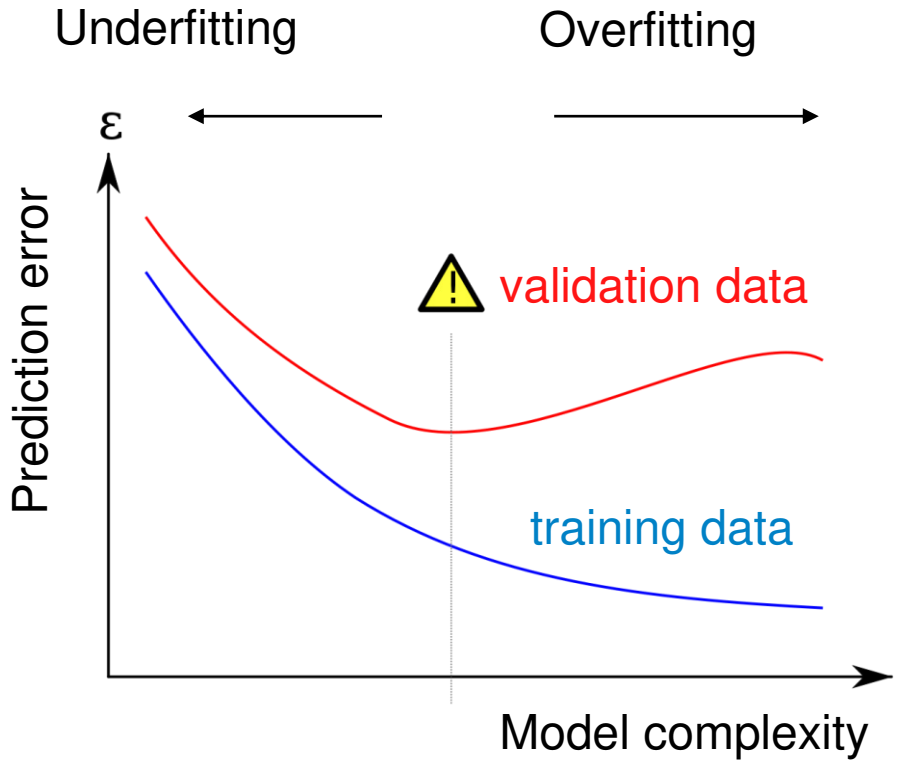
Overfitting & Underfitting



A well-fit model



An overfit model



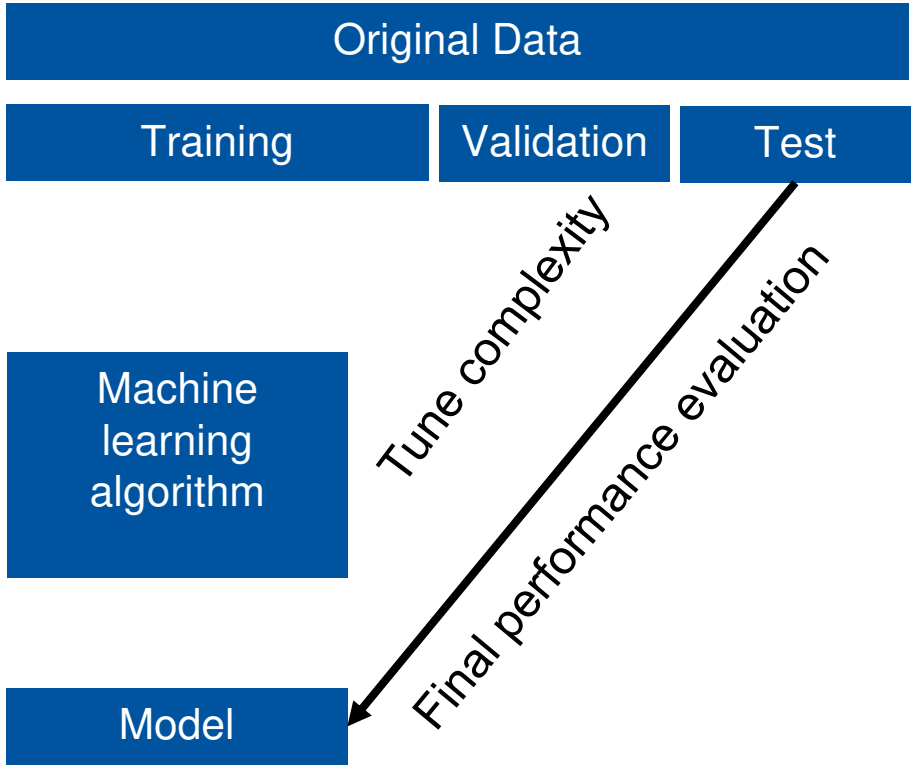
B. W. Schuller. “Deep Learning for Signal Analysis”, Deep Learning Summer School 2018.

Training, Test and Validation Data

- Training: minimize error between model and data
  - Adjust weights and biases of neurons
  - Use of gradient-based updates beneficial

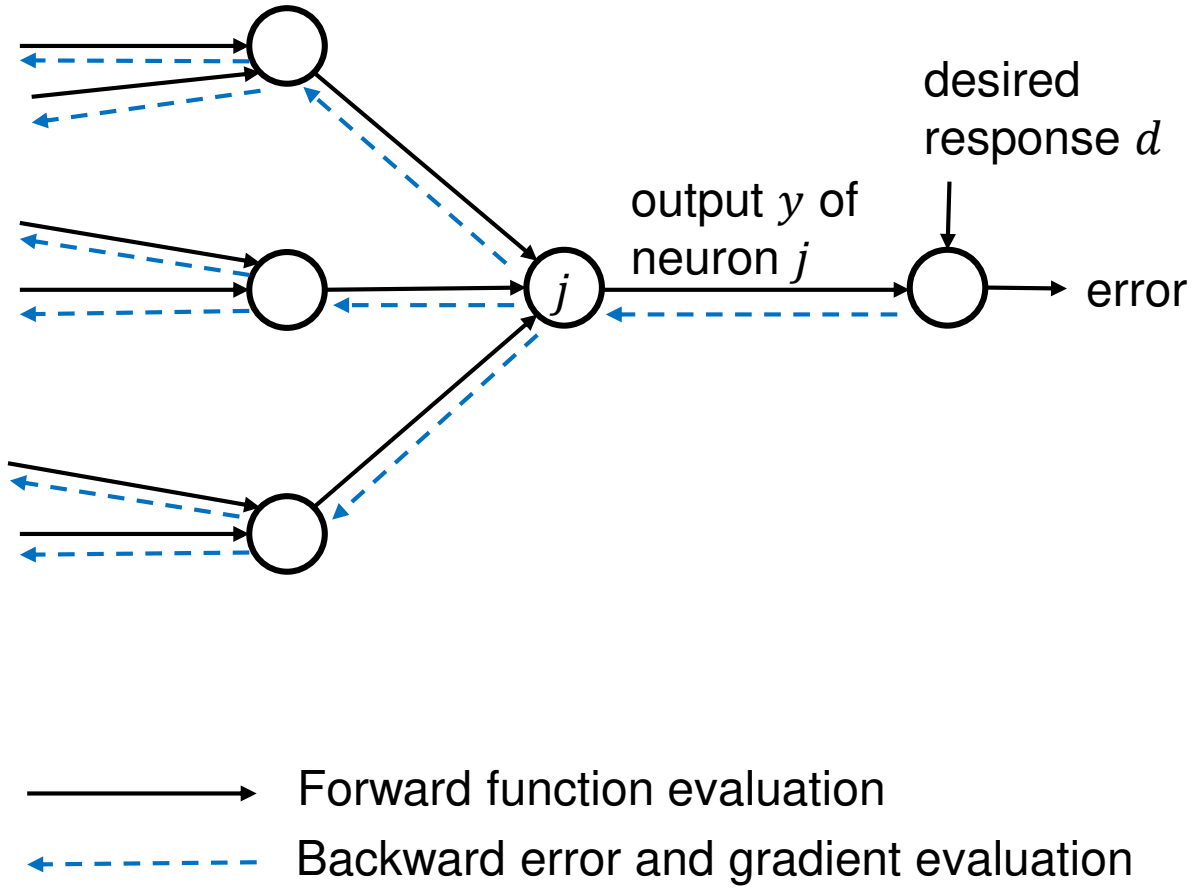
- Divide data randomly into 3 sets: training, validation and test
  - Predefined ratio, e.g., 70%:15%:15%
  - Fit weights and bias with the training set
  - Choose network complexity with the validation set
  - Evaluate performance with the test set

- Different language in ML vs. optimization
  - Training ↔ regression
  - loss function ↔ objective function
  - weights, biases ↔ decision variables (or parameters)
  - backpropagation algorithm ↔ gradient based method



Backpropagation Algorithm for Training ANNs

- Standard method for training
- Two passes of the ANN
  1. Evaluate functions in forward pass
  2. Evaluate gradients in backward pass
- Need chain rule: weights of neuron affect subsequent neurons

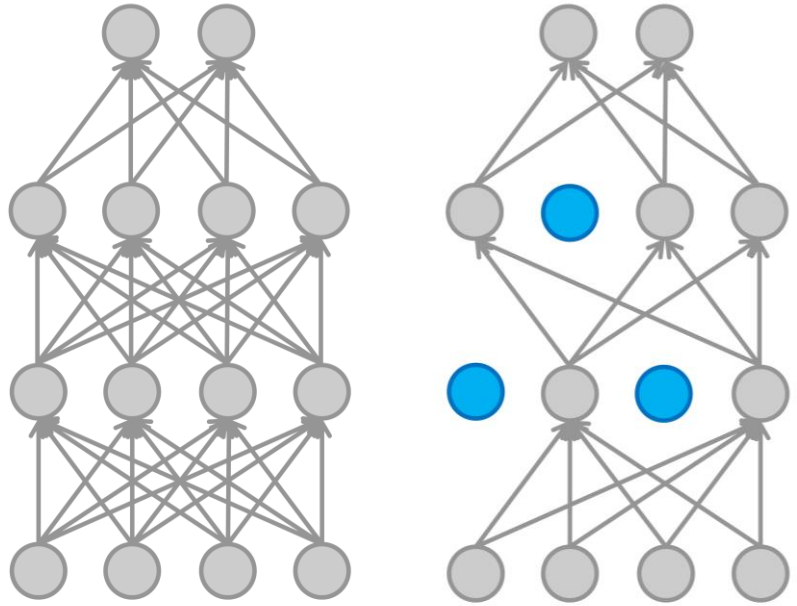


Regularization in Training

- Large neural networks include many parameters
  → risk of overfitting

- Regularization mitigates risk of overfitting, e.g.,
  - **Weight decay**: penalize large weights
  - **Dropout**: repeat training iterations with randomly dropped units

Dropout



B. W. Schuller. “Deep Learning for Signal Analysis”, Deep Learning Summer School 2018.

Check Yourself

- How does the backpropagation algorithm work in principle?
- How is the available data split? What is done with each of the chunks?
- What regularization is used to avoid overfitting?

Applied Numerical Optimization

Prof. Alexander Mitsos, Ph.D.

Deterministic global optimization with ANN embedded

Aachener Verfahrenstechnik | RWTH AACHEN UNIVERSITY

Recap: Full-Space vs. Reduced-Space[1-5]

$\mathbf{x}$ degrees of freedom
$\boldsymbol{z}$ state variables

Full-Space (FS)

$$\min_{\mathbf{x},\boldsymbol{z}} \; f(\mathbf{x},\boldsymbol{z})$$
$$\text{s.t.} \; \mathbf{g}(\mathbf{x},\boldsymbol{z}) \leq \mathbf{0}$$
$$\mathbf{h}(\mathbf{x},\boldsymbol{z}) = \mathbf{0}$$

Total dim: $\dim(\mathbf{x}) + \dim(\boldsymbol{z})$
with $\dim(\mathbf{x}) \ll \dim(\boldsymbol{z})$

Solve $\mathbf{h}(\mathbf{x},\boldsymbol{z}) = \mathbf{0}$ for $\mathbf{z}$
(if possible)

Reduced-Space (RS)

$$\min_{\mathbf{x}} \; \tilde{f}(\mathbf{x})$$
$$\text{s.t.} \; \tilde{\mathbf{g}}(\mathbf{x}) \leq \mathbf{0}$$

Total dim: $\dim(\mathbf{x})$

[1] Epperly & Pistikopoulos, JOGO, 11(3), 287-311 (1997) [2] Byrne & Bogle, Ind. Eng. Chem. Res, 39(11), 4296-4301 (2000). [3] Mitsos, Chachuat & Barton, SIOPT, 20(2), 573-601 (2009) [4] Bongartz, & Mitsos, JOGO, 69(4), 761-796 (2017) [5] Bongartz and Mitsos, JOGO, 69(4), 761-796 (2018) [6] Schweidtmann, A. M., & Mitsos, A..JOTA, 180(3), 925-948 (2019).

Reduced Space Formulation with ANNs-Embedded

Full-space

$$\min_{x,z} y_{\text{ANN}}$$

All model variables

s.t.

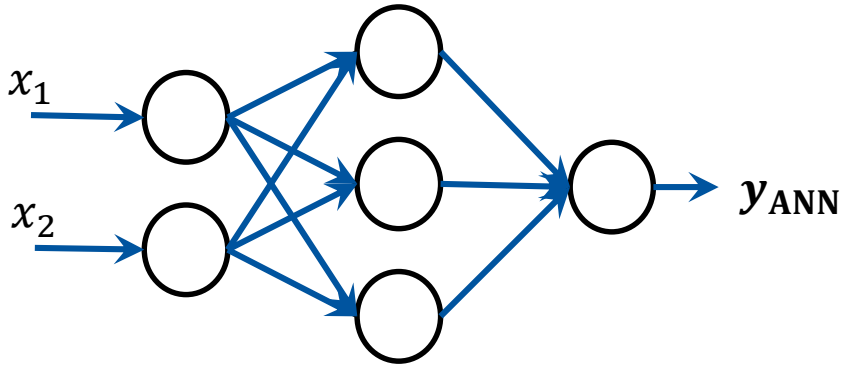
$$y_{\text{ANN}} = \left(z_1^{(3)} + 1\right) \cdot \frac{y^U - y^L}{2} + y^L$$

$$z_1^{(3)} = \sum_1^3 w_{j,1}^{(2)} z_j^{(2)} + b_1^{(2)}$$

$$z_j^{(2)} = \tanh\left(\sum_1^3 w_{j,i}^{(1)} z_j^{(1)} + b_i^{(1)}\right)$$

ECs describing the ANN

$$z_j^{(1)} = \frac{x_j - x_j^L}{x_j^U - x_j^L} \cdot 2 - 1$$



Reduced-space

$$\min_{x} f_{\text{ANN}}(x)$$

Sequential evaluation of (feed-forward) ANN

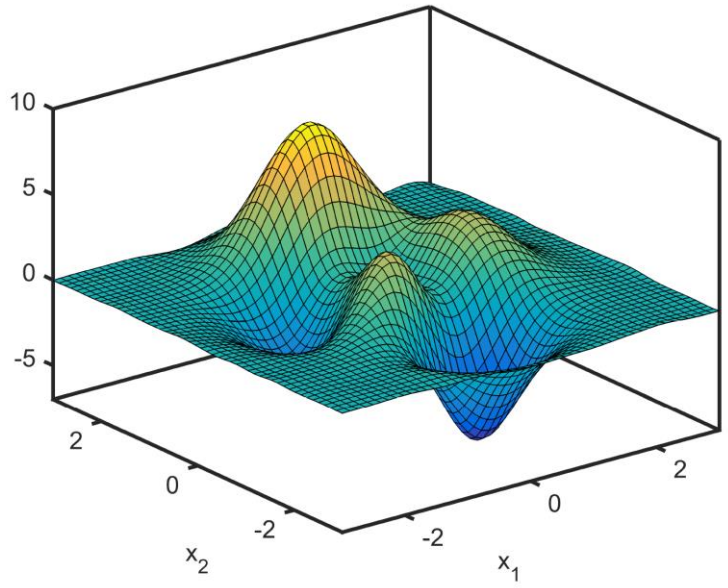
Only DoF

☺ Suitable for standard solvers
☺ Simple functions
☹ Large-scale NLPs

☺ Small scale: $\dim(z) \gg \dim(x)$
☹ Relaxations of f_{ANN} required

[1] A.M. Schweidtmann and A. Mitsos; Deterministic Global Optimization with Artificial Neural Networks Embedded; Journal of Optimization Theory and Applications (2019)

Example: The Peaks Function is learned by an ANN and Subsequently Optimized



- Train a feed-forward ANN
  - 500 Data points from peaks function
  - 1 hidden layer with 47 neurons
- Optimize the predictions of the ANN

Full space
101 optimization variables
99 equality constraints

Reduced-space
2 optimization variables
0 equality constraints

161.35 seconds**	MAiNGO	0.50 seconds**
	Speedup by factor about 300	
21.08 to >10^5 seconds*	GAMS BARON	-

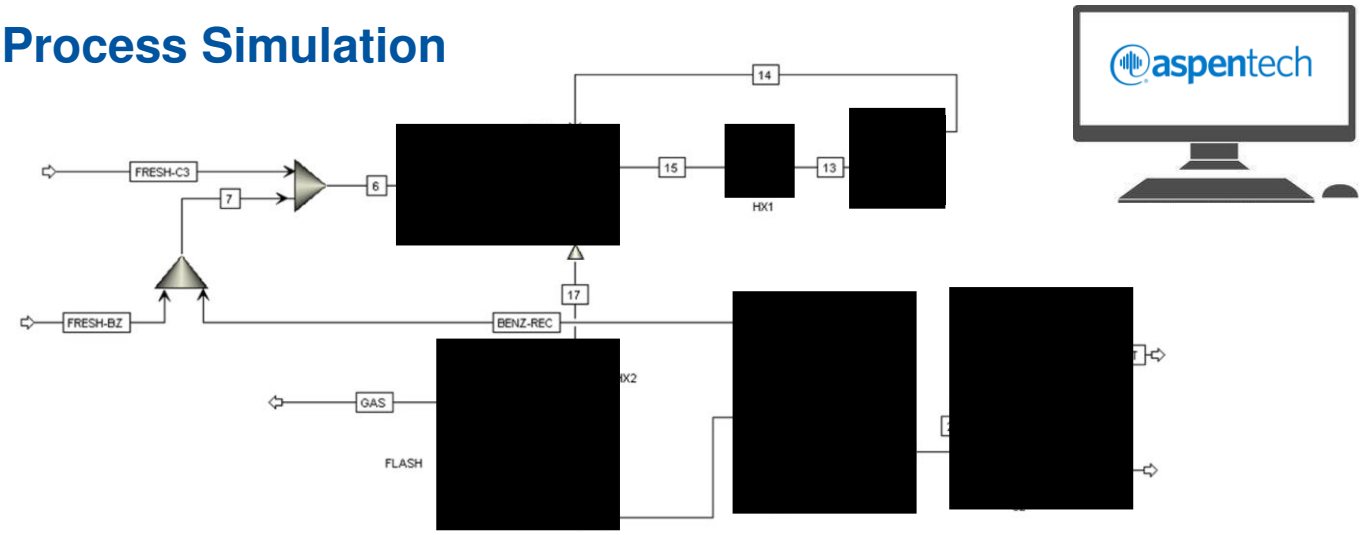
*using different algebraic formulations of tanh activation function, ** using the implemented envelope of tanh

[1] A.M. Schweidtmann and A. Mitsos; Deterministic Global Optimization with Artificial Neural Networks Embedded; Journal of Optimization Theory and Applications (2019)

Example: Hybrid Model for Chemical Process Optimization

1. Decomposition
2. Data generation
3. Training
4. Hybrid model
5. Optimization

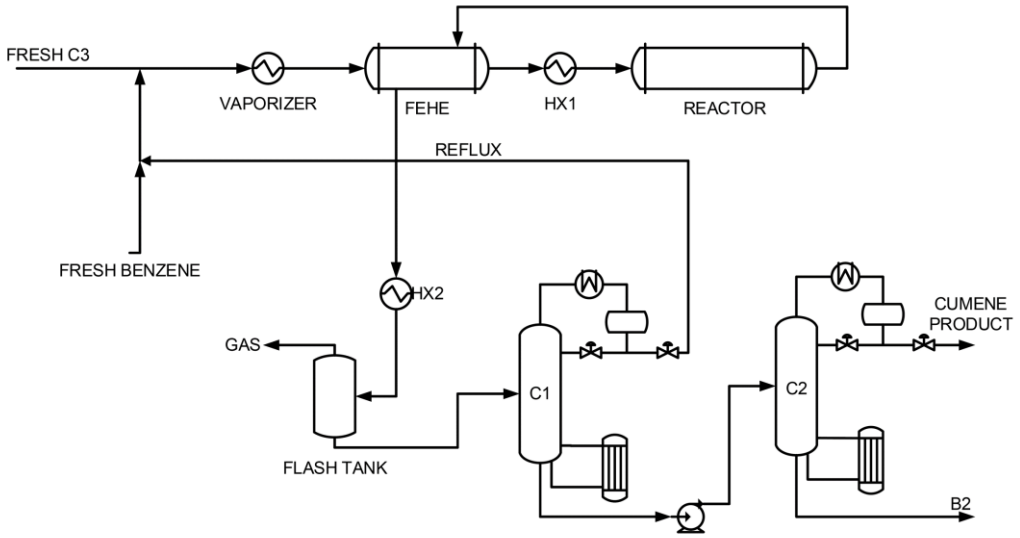
Process Simulation

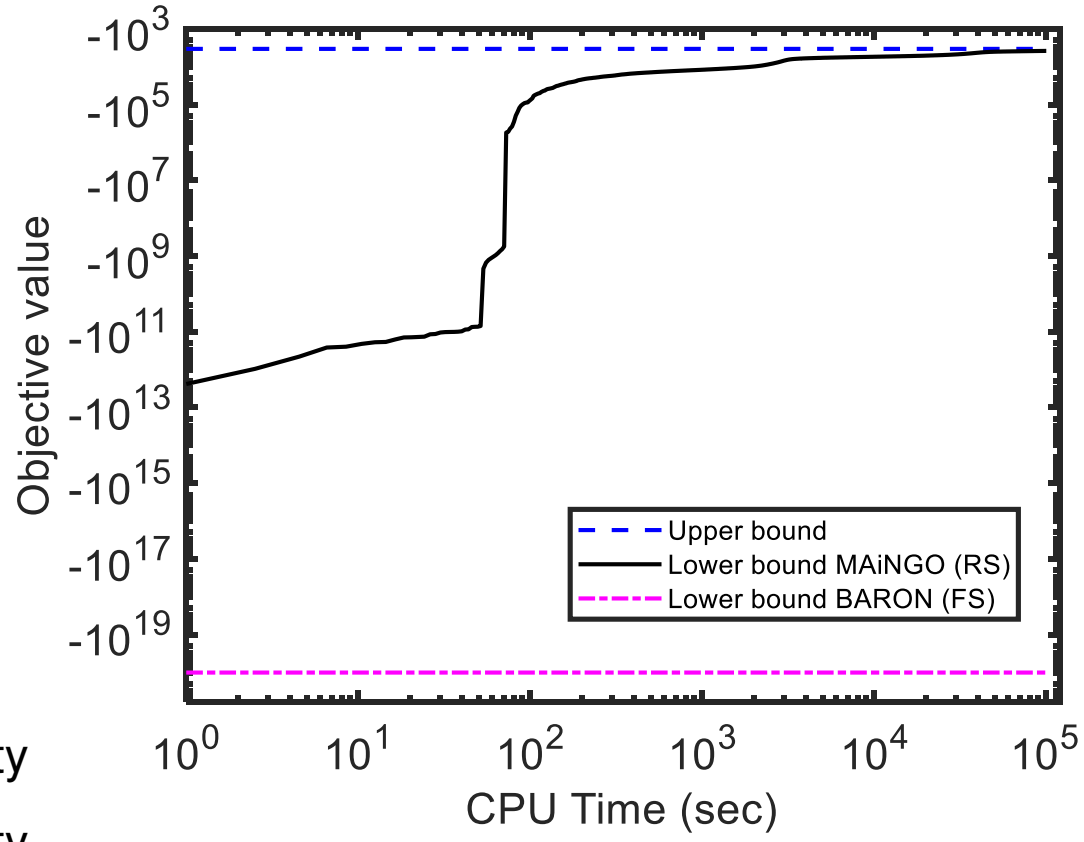


- Not single black box
- Hybrid process flowsheet including 14 ANNs
- ANNs connected via balance equations

[1] A.M. Schweidtmann and A. Mitsos; Deterministic Global Optimization with Artificial Neural Networks Embedded; Journal of Optimization Theory and Applications (2019)

Example: Hybrid Model for Chemical Process Optimization – Results





Problem size:

Full-space (FS): 794 variables, 789 equality, 1 inequality

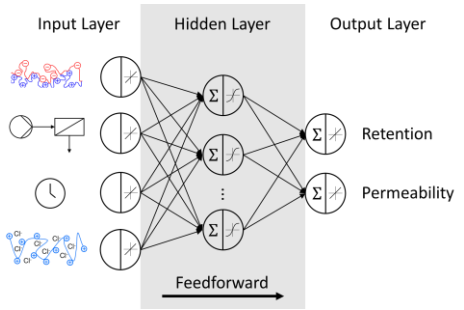
Reduced-space (RS): 5 variables, 0 equality, 1 inequality

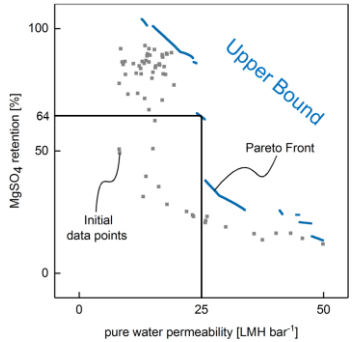
[1] A.M. Schweidtmann and A. Mitsos; Deterministic Global Optimization with Artificial Neural Networks Embedded; Journal of Optimization Theory and Applications (2019)

More Examples – All Applications Use Our In-house Solver MAiNGO[5]

Rational design of ion separation membranes[1]

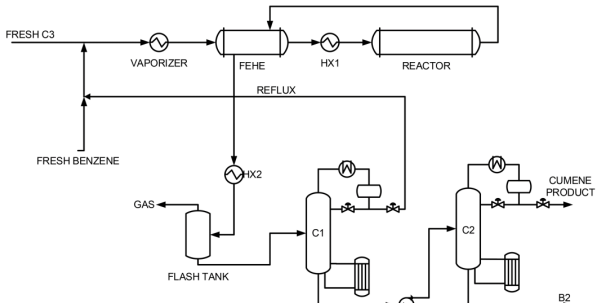
- Learn membrane properties as a function of fabrication parameters from experimental data
- Multi-objective optimization to identify Pareto optimal membranes (retention and permeability)

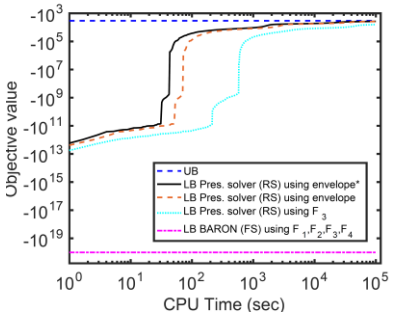




Cumene process optimization[2,3]

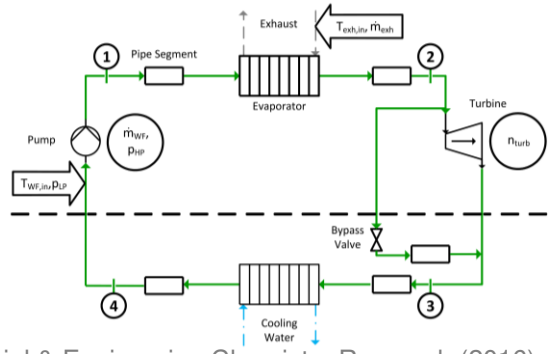
- Unit operations simulated in AspenPlus
- Individual learning of unit operation
- Aggregation to hybrid process model

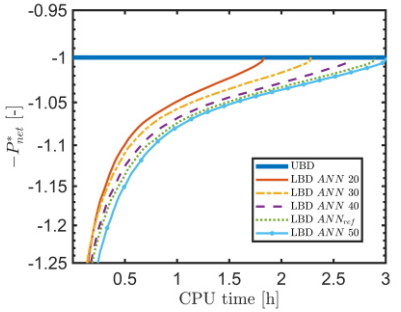




Learning single species thermodynamic properties[4]

- Organic Rankine cycle process optimization
- Learn thermodynamic properties from simulated data
- Faster optimization compared to Helmholtz EOS





[1] Rall et al., Journal of Membrane Science (2019) [2] Schweidtmann & Mitsos, JOTA (2019) [3] Schultz & Jorge, Industrial & Engineering Chemistry Research (2016)
[4] Schweidtmann, Huster, & Mitsos, Computers & Chemical Engineering (2019) [5] Bongartz et al., MAiNGO technical report (2018)

Check Yourself

- What is reduced space formulation for optimization with ANN embedded?
- Why is it beneficial to use reduced space formulation/ MAiNGO for global optimization with ANNs embedded?

Applied Numerical Optimization

Prof. Alexander Mitsos, Ph.D.

Gaussian Processes (Kriging) and Bayesian Optimization

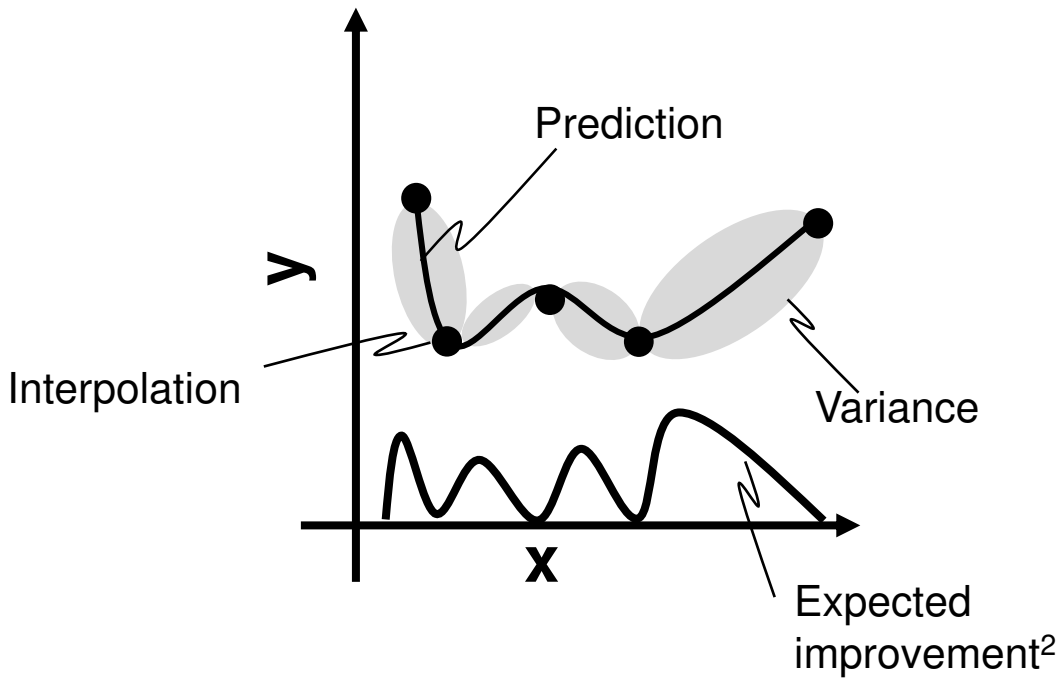
Aachener Verfahrenstechnik | RWTH AACHEN UNIVERSITY

Motivation

- Most data-driven models provide only a prediction
  - No information about uncertainty of predictions

- Gaussian processes (GPs) a.k.a. Kriging[1]
  - Generalization of multivariate Gaussian distributions
  - Nonlinear interpolation method
  - Provide prediction and variance
  - Well-suited for sparse data
  - Can be used in Bayesian optimization[2]

- Many applications for model-free optimization of expensive-to-evaluate functions
  - Process simulations[3], NMPC for bioprocesses[4], hardware-in-the-loop[5], hyperparameter optimization in ML[6]



[1] Krige, D. G. (1951). Journal of the Southern African Institute of Mining and Metallurgy, 52(6), 119-139. [2] Jones, Schonlau, & Welch, *Journal of Global Optimization (1998). 13(4). 455-492.* [3] Helmdach, D., Yaseneva, P., Heer, P. K., Schweidtmann, A. M., & Lapkin, A. A. (2017). ChemSusChem, 10(18), 3632-3643. [4] Bradford, E., Schweidtmann, A. M., Zhang, D., Jing, K., & del Rio-Chanona, E. A. (2018). Computers & Chemical Engineering, 118, 143-158. [5] Schweidtmann, A. M., Clayton, A. D., Holmes, N., Bradford, E., Bourne, R. A., & Lapkin, A. A. (2018). Chemical Engineering Journal, 352, 277-282. [6] Snoek, J., Larochelle, H., & Adams, R. P. (2012). In Advances in neural information processing systems (pp. 2951-2959).

Optimization With Gaussian Processes Embedded is Difficult

Full-Space

$$\min_{x_*, \bar{y}_*, K_*, k, r \ldots} \bar{y}_*$$

$$\text{s.t. } \bar{y}_* = K_* \cdot K^{-1} \cdot y$$

Training targets

$$K = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \cdots & k(x_1, x_N) \\ k(x_2, x_1) & k(x_2, x_2) & \ldots & k(x_2, x_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n, x_1) & k(x_n, x_2) & \cdots & k(x_N, x_N) \end{bmatrix}$$

Covariance matrix (N x N)

$$K_* = [k(x_*, x_1) \quad k(x_*, x_2) \quad \cdots \quad k(x_*, x_N)]$$

$$k(x_i, x_j) = \sigma_f^2 e^{\left(-\frac{r_{ij}}{l}\right)}, k \in [k^l, k^u]$$

Covariance function

$$r_{ij} = \| x_i - x_j \| \qquad r_{ij} \in [{r_{ij}}^l, {r_{ij}}^u] \qquad i, j \in [1, N]$$

$$x_* \in [x^l, x^u] \subset R^D$$

Euclidian distance

D: Dimension of training points

N: Number of training points

Total dim:

Variables: $\mathcal{O}(N + D)$

Equalities: $\mathcal{O}(N + D)$

usually $N \gg D$

→ Gaussian processes lead to large-scale nonlinear problems in the full-space formulation

Gaussian Processes in a Reduced-Space

Full-Space

$$\min_{\boldsymbol{x}_*, \bar{y}_*, \boldsymbol{K}_*, k, r \dots} \bar{y}_*$$

s.t. $\bar{y}_* = \boldsymbol{K}_* \cdot \boldsymbol{K}^{-1} \cdot \boldsymbol{y}$

$$\boldsymbol{K} = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \cdots & k(x_1, x_N) \\ k(x_2, x_1) & k(x_2, x_2) & \dots & k(x_2, x_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n, x_1) & k(x_n, x_2) & \cdots & k(x_N, x_N) \end{bmatrix}$$

$$\boldsymbol{K}_* = [k(\boldsymbol{x}_*, \boldsymbol{x}_1) \quad k(\boldsymbol{x}_*, \boldsymbol{x}_2) \quad \cdots \quad k(\boldsymbol{x}_*, \boldsymbol{x}_N)]$$

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sigma_f^2 e^{\left(-\frac{\boldsymbol{r}_{ij}}{l}\right)}, k \in [k^l, k^u]$$

$$r_{ij} = \| \boldsymbol{x}_i - \boldsymbol{x}_j \| \qquad r_{ij} \in [r_{ij}{}^l, r_{ij}{}^u] \qquad i, j \in [1, N]$$

$$\boldsymbol{x}_* \in [\boldsymbol{x}^l, \boldsymbol{x}^u] \subset R^D$$

Total dim: # Variables: $\mathcal{O}(N + D)$; # Equalities: $\mathcal{O}(N + D)$

- Solve equalities for y_{out}
- Propagate McCormick relaxations and (sub) gradient through code

Reduced-space

Sequential evaluation of Gaussian process

$$\min_{\boldsymbol{x}_*} \mathcal{GP}(\boldsymbol{x}_*)$$

s.t. $\boldsymbol{x}_* \in [\boldsymbol{x}^l, \boldsymbol{x}^u] \subset \mathbb{R}^D$

Only DoF

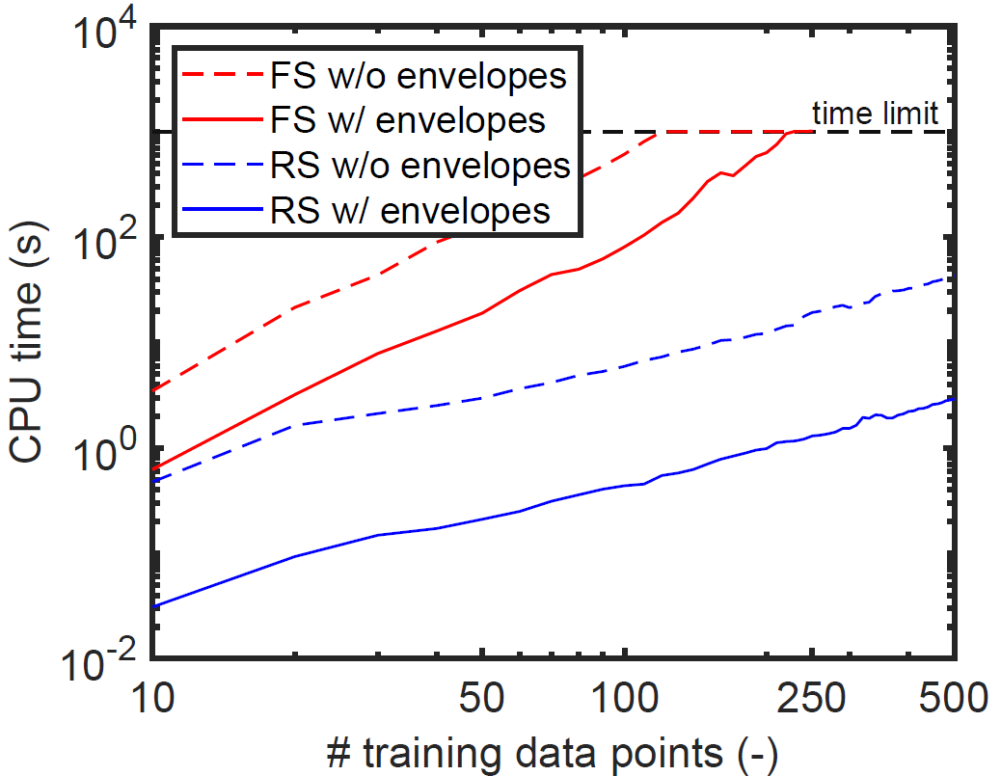
Total dim:
Variables: D
Equalities: 0

D: Dimension of training points N: Number of training points

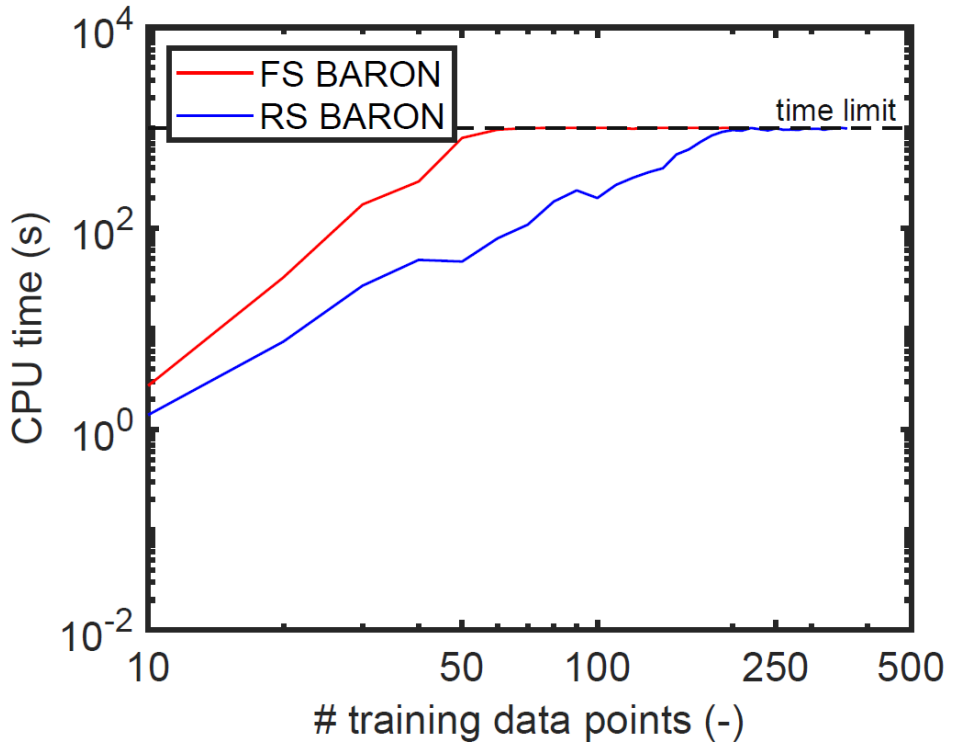
→ The reduced-space formulation reduces problem size drastically

Scaling of Computational Performance with Training Set Size on Peaks Function

MAiNGO



BARON



→ The reduced-space formulation outperforms the full-space formulation for Gaussian processes

[1] Schweidtmann, A. M., Bongartz, D., Grothe, D., Kerkenhoff, T., Lin, X., Najman, J., & Mitsos, A. (2020). Global Optimization of Gaussian processes. arXiv preprint arXiv:2005.10902.

Mixed-integer Bayesian Optimization of Membrane Synthesis[1]

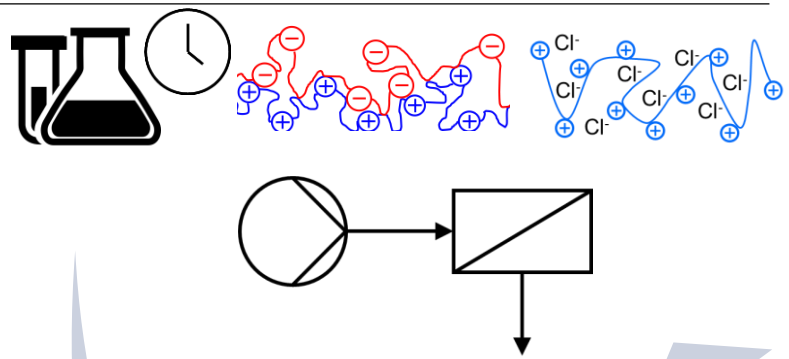
Mixed-Integer optimization problem:

$$\max \text{Expected Improvement}_{\text{Na}_2\text{SO}^4\text{ Retention}}$$

$$\text{s.t. } x_{LB,i} \leq x_i \leq x_{UB,i} \quad i = 1,..,3$$

2 continuous variables:
sodium chloride concentration: $c_{NaCl} \in [0,0.5]\text{gL}^{-1}$
deposited polyelectrolyte mass: $\text{m}_{\text{PE}} \in [0,5]\text{gm}^{-2}$

1 integer variable:
number of layers: $N_{Layers} \in \{1,2,3 \dots 10\}$



Global solution found within 7 minutes

- $c_{NaCl} = 0.362\text{gL}^{-1}$, $\text{m}_{\text{PE}} = 0\ \text{gm}^{-2}$, $N_{Layers} = 4$
- Expected retention 85.32 with standard deviation $\sigma = 14.8$

Data from collaboration with

DWI Leibniz-Institut für Interaktive Materialien

[1] Schweidtmann, Bongartz, Grothe, Kerkenhoff, Lin, Najman, Mitsos (2020). https://arxiv.org/abs/2005.10902

Check Yourself

- What are GP?
- What is Bayesian optimization?
- Why is it beneficial to use MAiNGO for global optimization with ANNs/GP embedded?