TUTORIAL MACHINE LEARNING IN SOLID MECHANICS

TECHNISCHE
UNIVERSITÄT
DARMSTADT

TASK 1: FEED-FORWARD NEURAL NETWORKS

Fabian Roth

TUTORIAL MACHINE LEARNING IN SOLID MECHANICS

SECTIONS

TECHNISCHE
UNIVERSITÄT
DARMSTADT

1 Hyperparameter sweep

2 Input convex neural networks

3 Trainable custom layer

4 Sobolev training

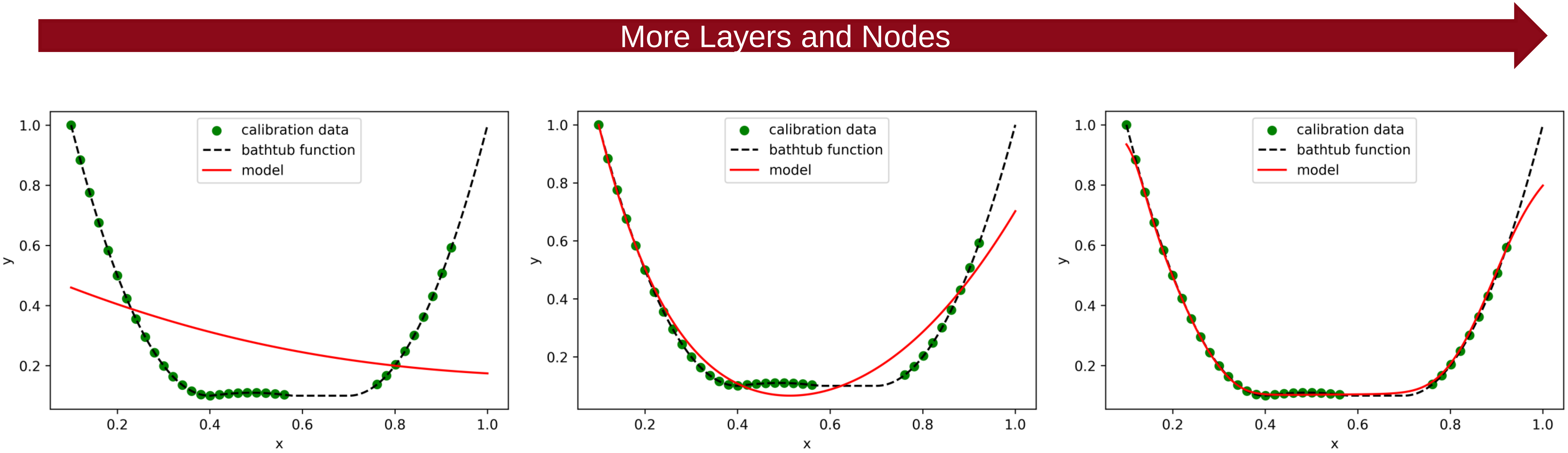
TECHNISCHE
UNIVERSITÄT
DARMSTADT

TASK 1 – 1.1

HYPERPARAMETER SWEEP

HYPERPARAMETER SWEEP

MODEL COMPLEXITY

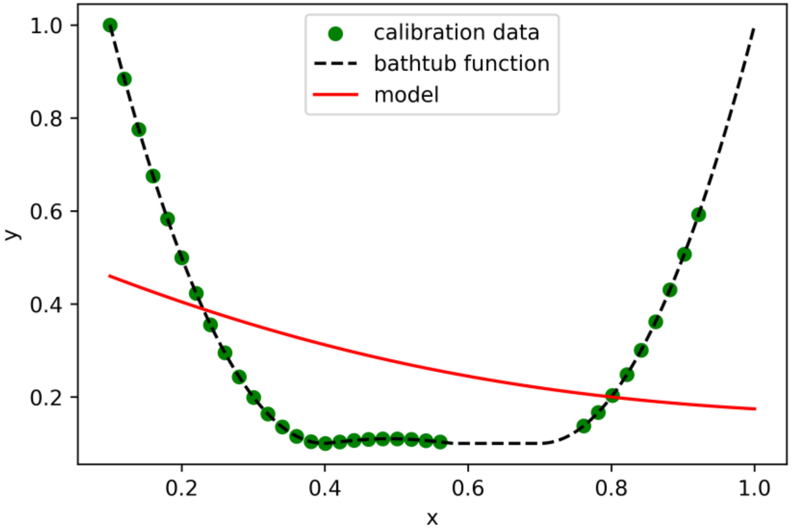


More Layers and Nodes

HYPERPARAMETER SWEEP

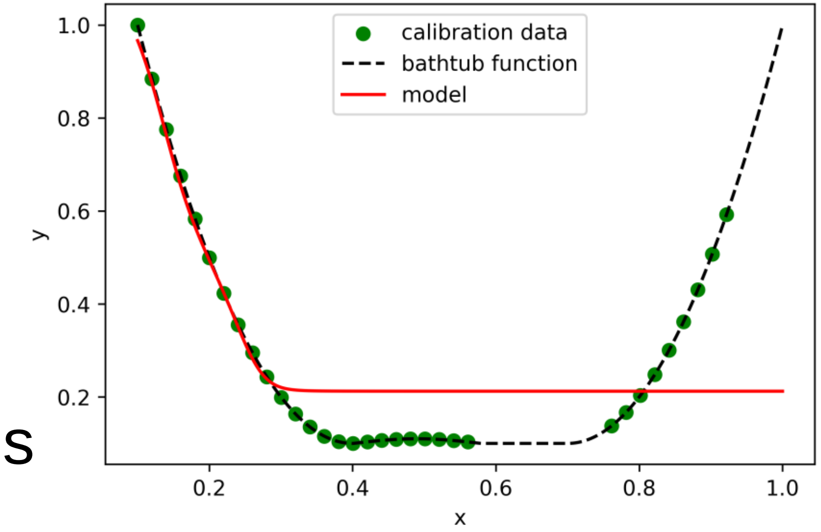
MODEL COMPLEXITY... IT’S NOT THAT SIMPLE

TECHNISCHE UNIVERSITÄT DARMSTADT



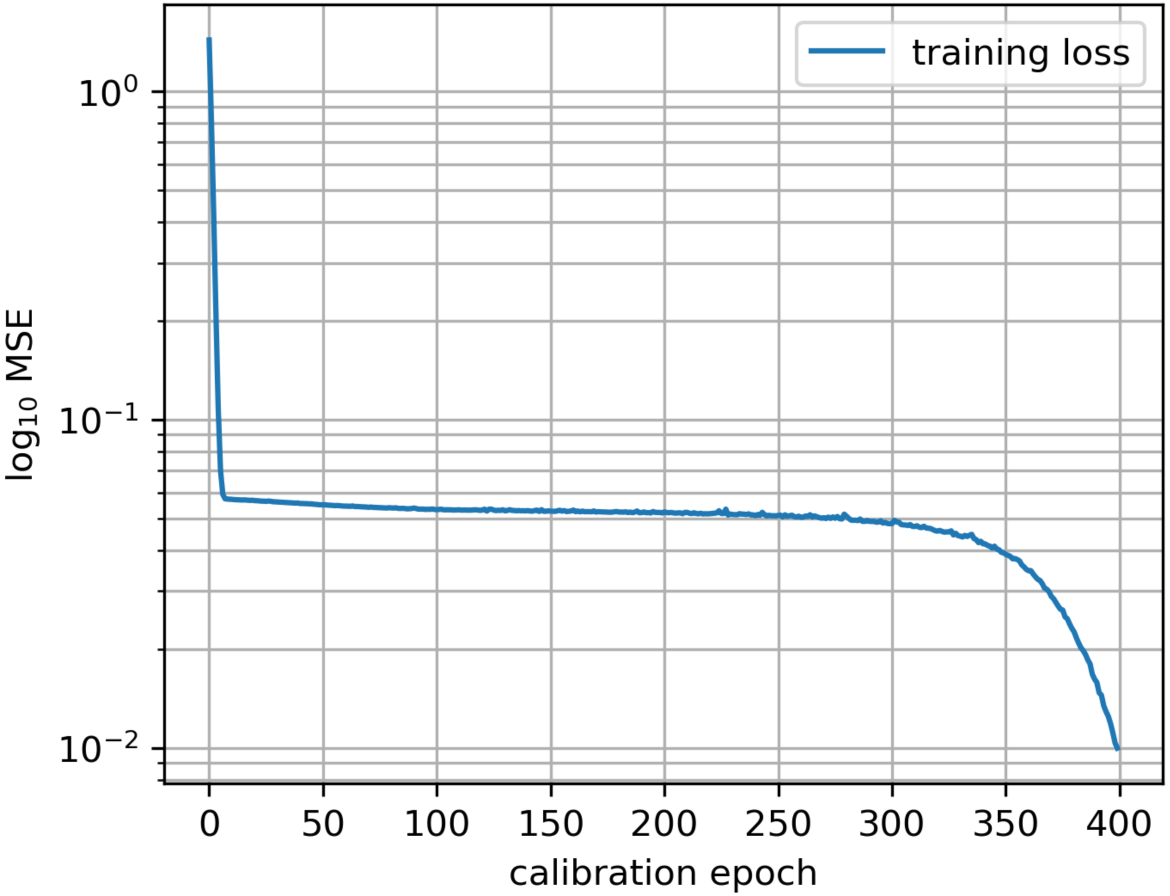
It‘s difficult to underfit on this simple data

Underfitting might have different causes



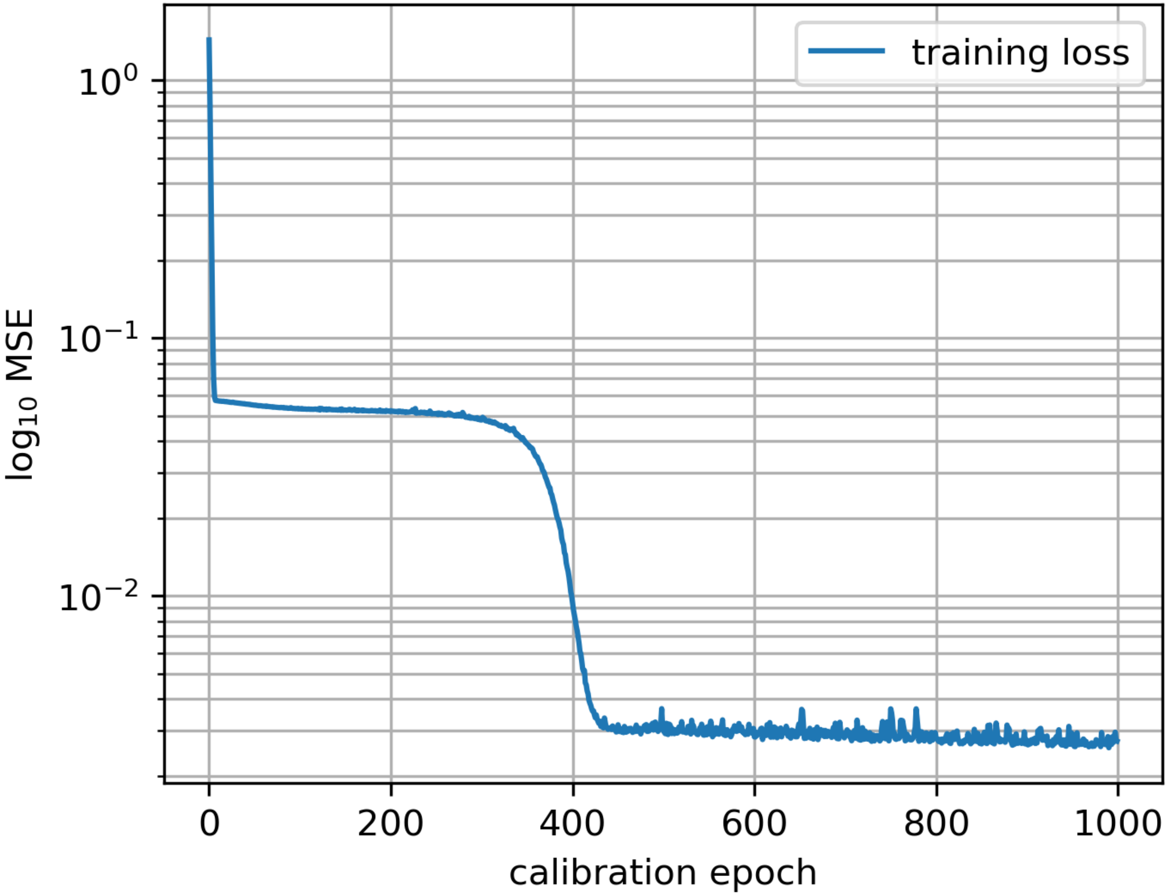
HYPERPARAMETER SWEEP

NUMBER OF EPOCHS



HYPERPARAMETER SWEEP

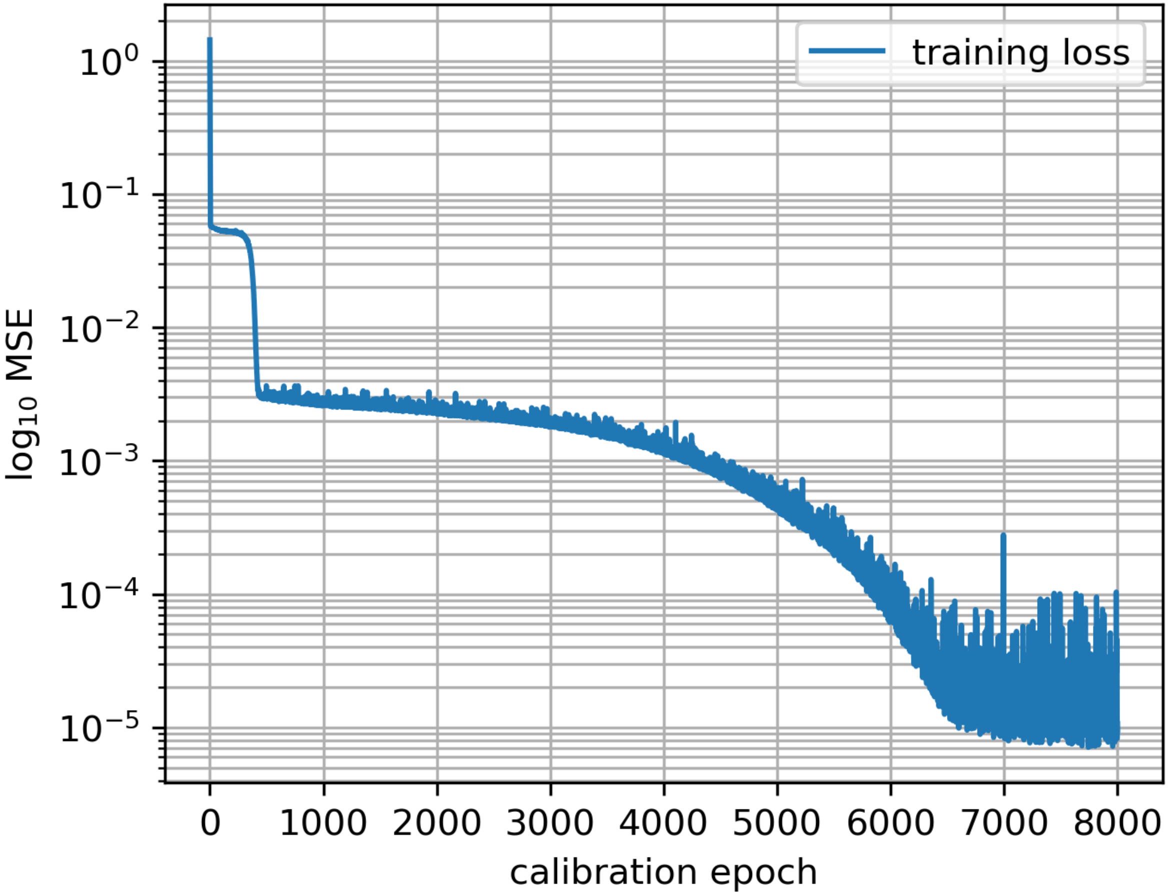
NUMBER OF EPOCHS



- Train until convergence

HYPERPARAMETER SWEEP

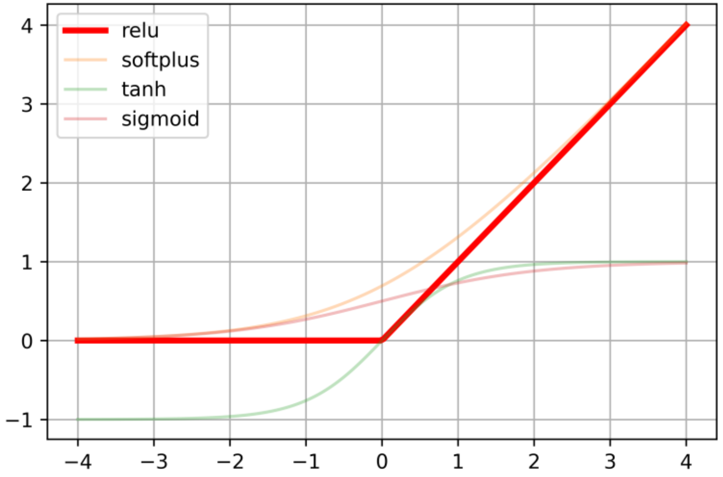
NUMBER OF EPOCHS

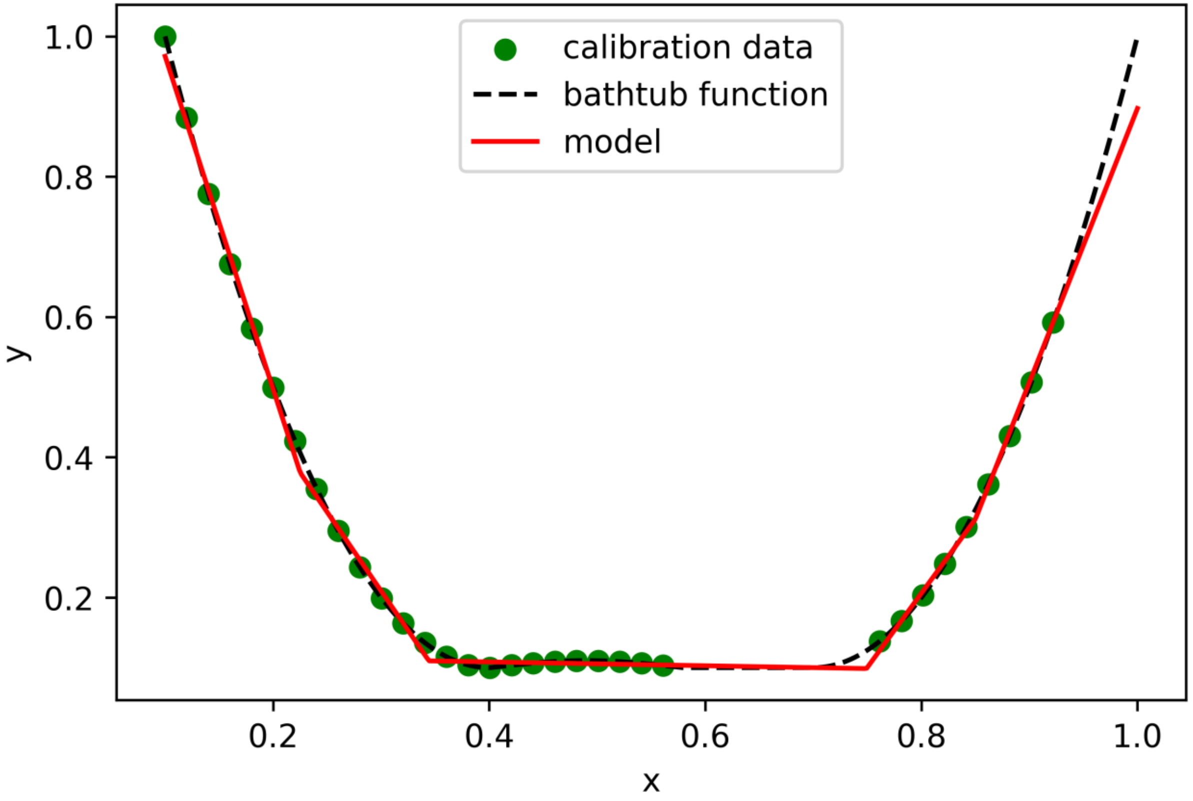


- Train until convergence
- “False“ convergence

HYPERPARAMETER SWEEP

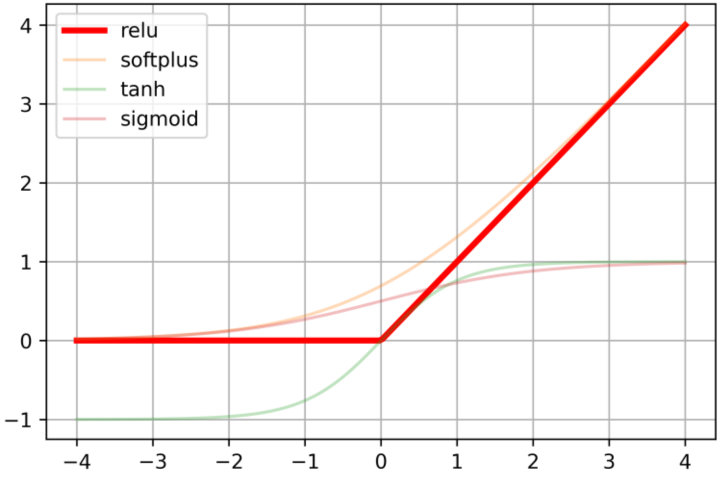
ACTIVATION FUNCTIONS - RELU

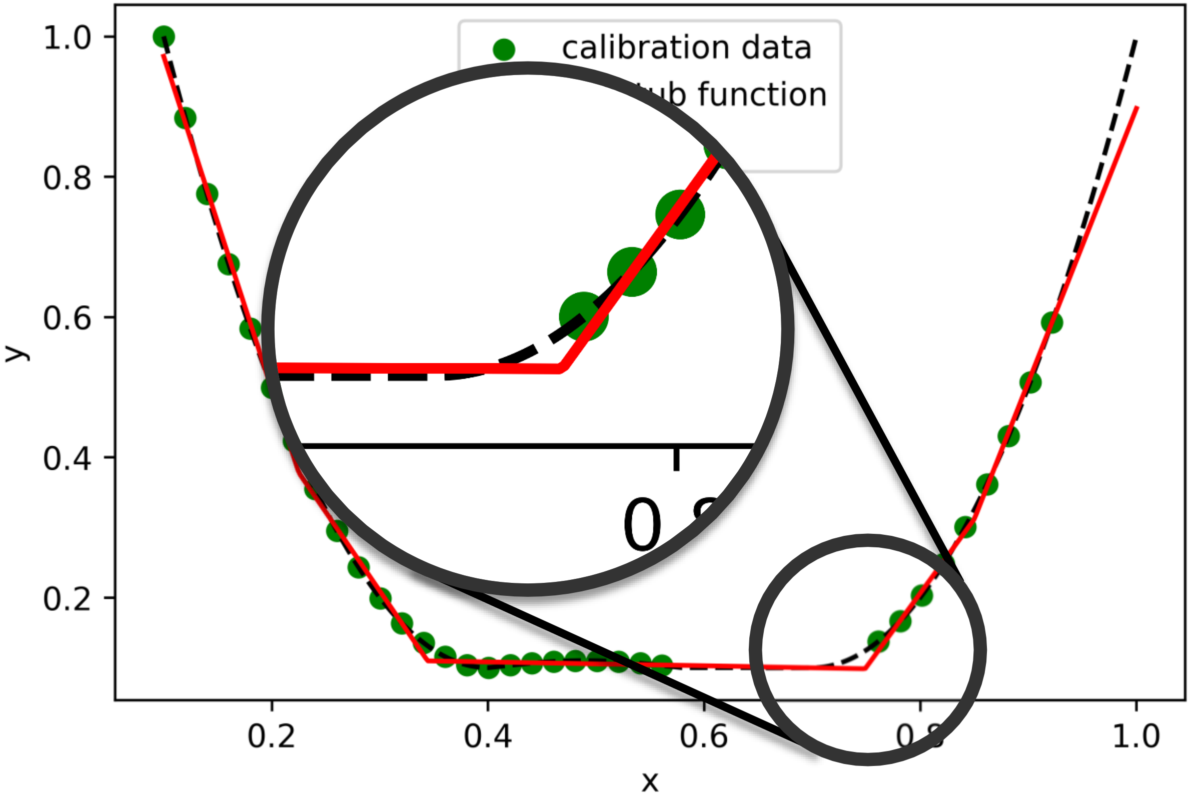




HYPERPARAMETER SWEEP

ACTIVATION FUNCTIONS - RELU

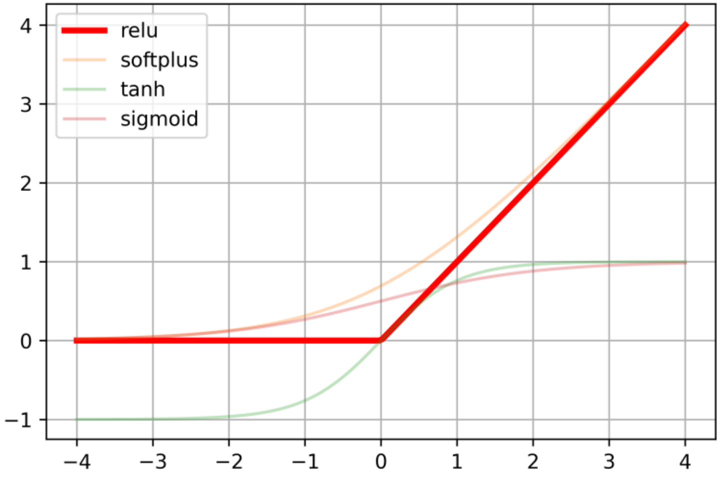


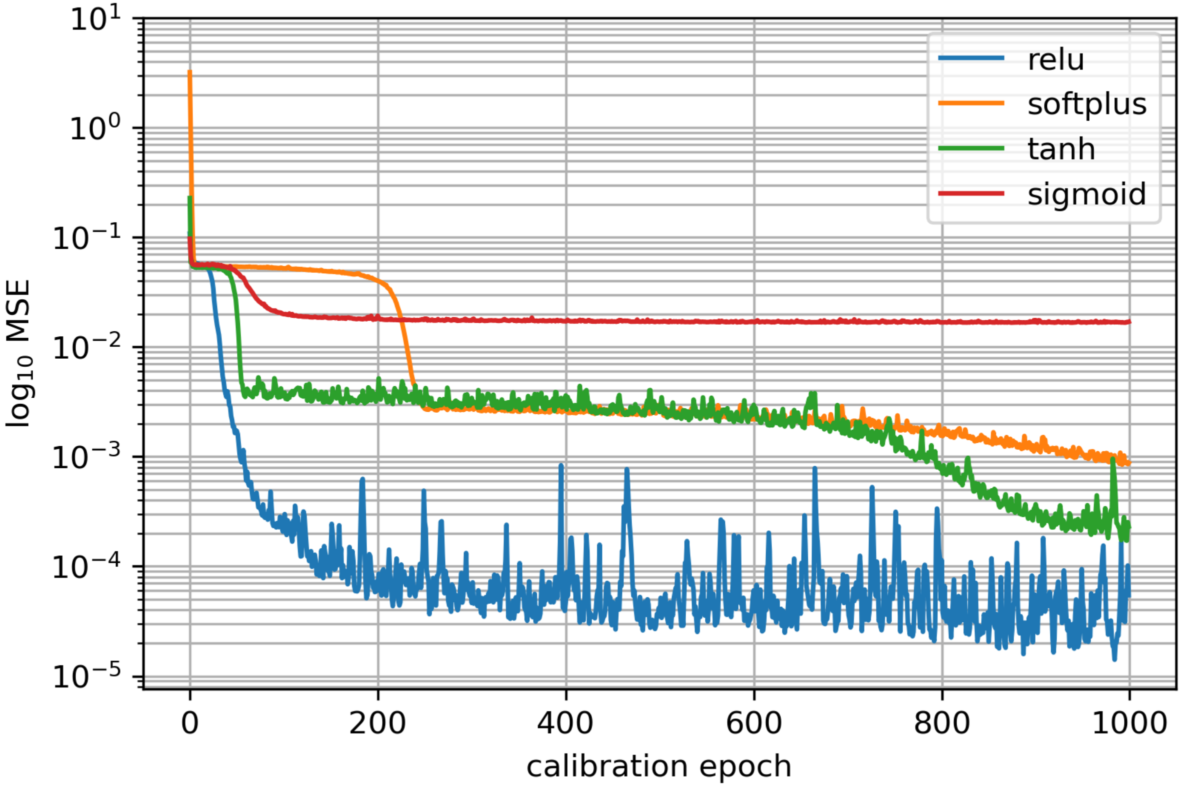


- Not continuously differentiable

HYPERPARAMETER SWEEP

ACTIVATION FUNCTIONS - RELU

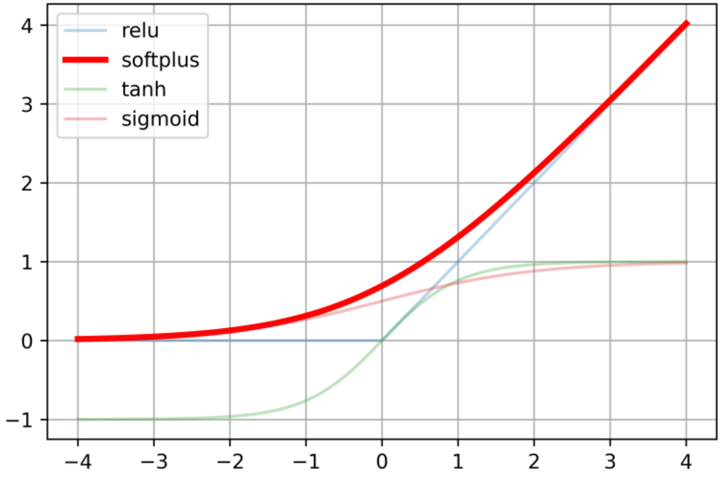


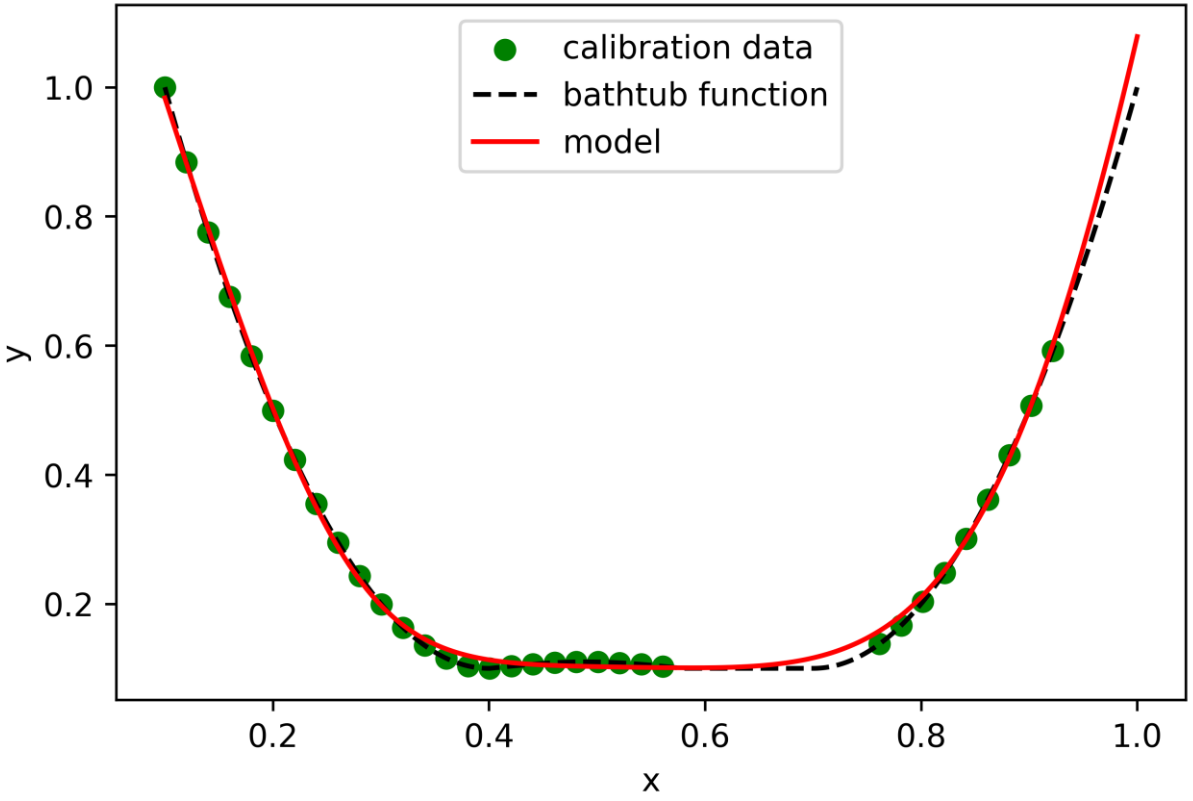


- Not continuously differentiable
- Fast convergence

HYPERPARAMETER SWEEP

ACTIVATION FUNCTIONS - SOFTPLUS

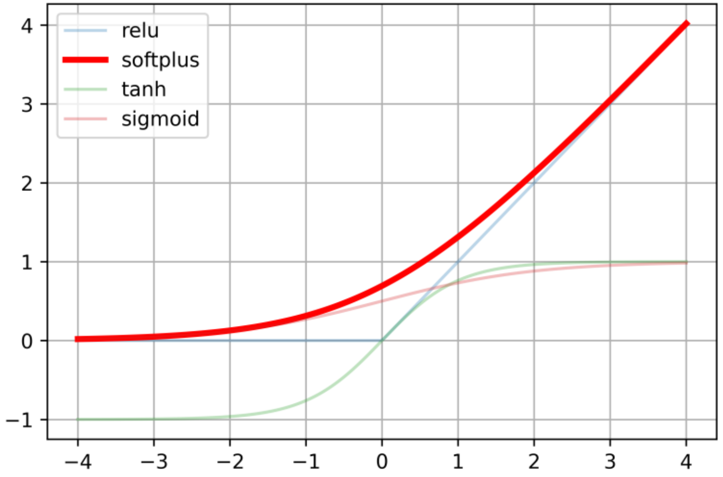


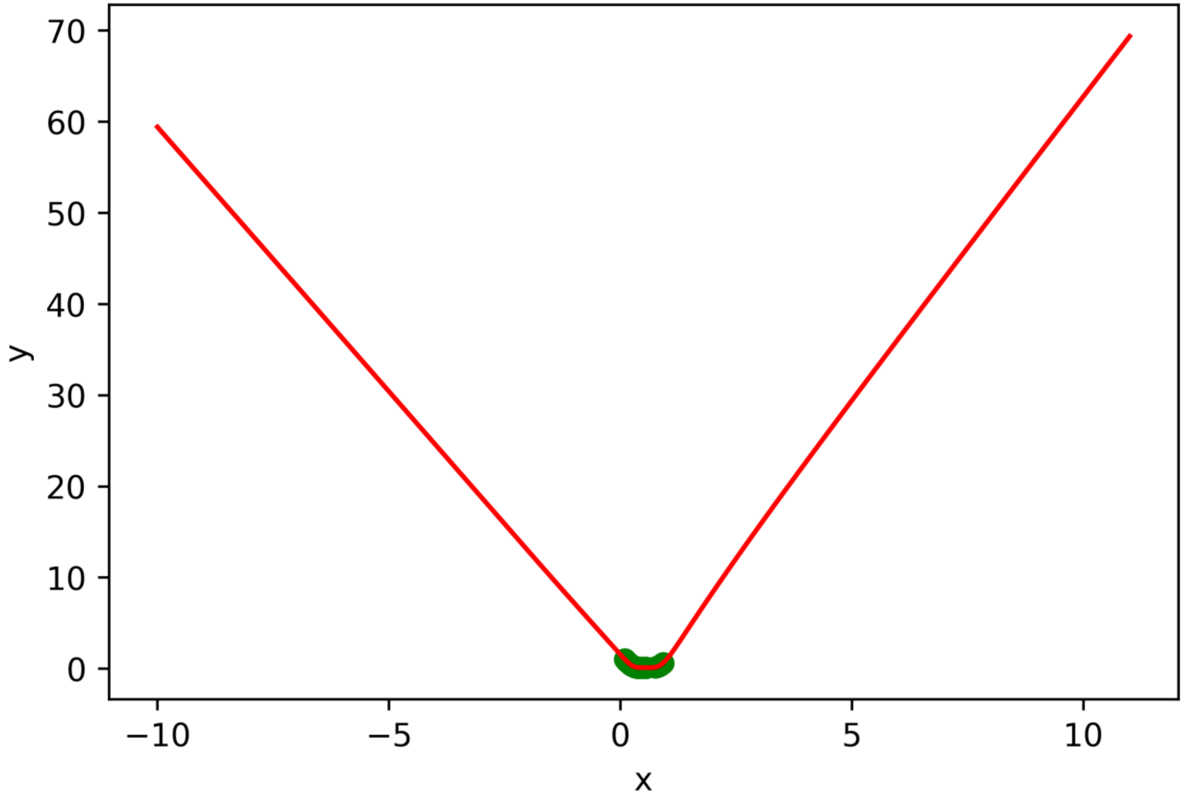


- Infinitely differentiable

HYPERPARAMETER SWEEP

ACTIVATION FUNCTIONS - SOFTPLUS

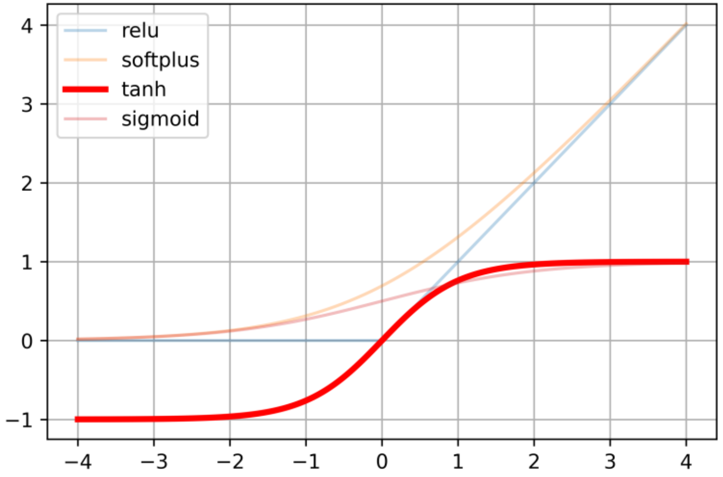




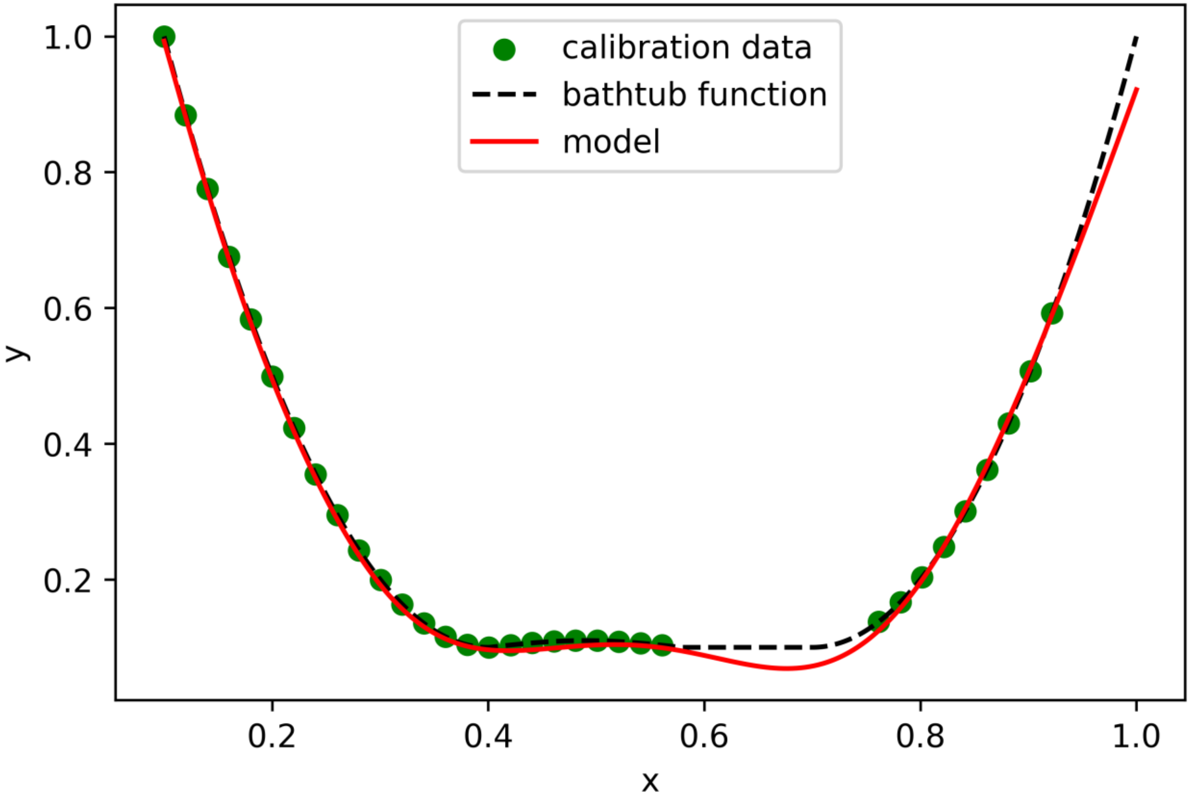
- Infinitely differentiable
- Unbounded

HYPERPARAMETER SWEEP

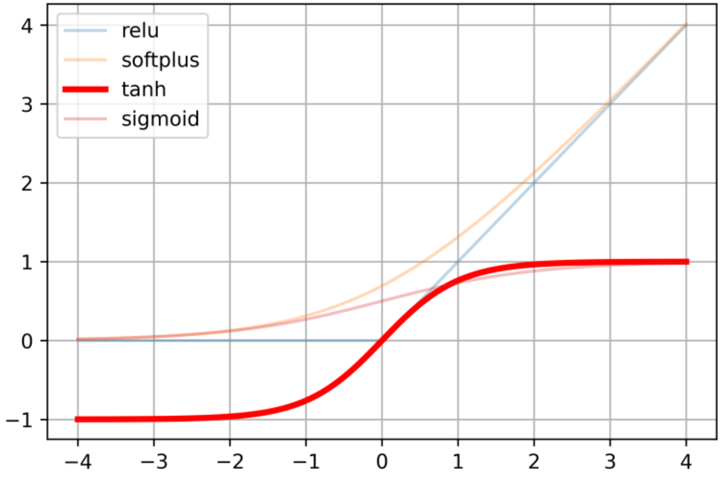
ACTIVATION FUNCTIONS - TANH

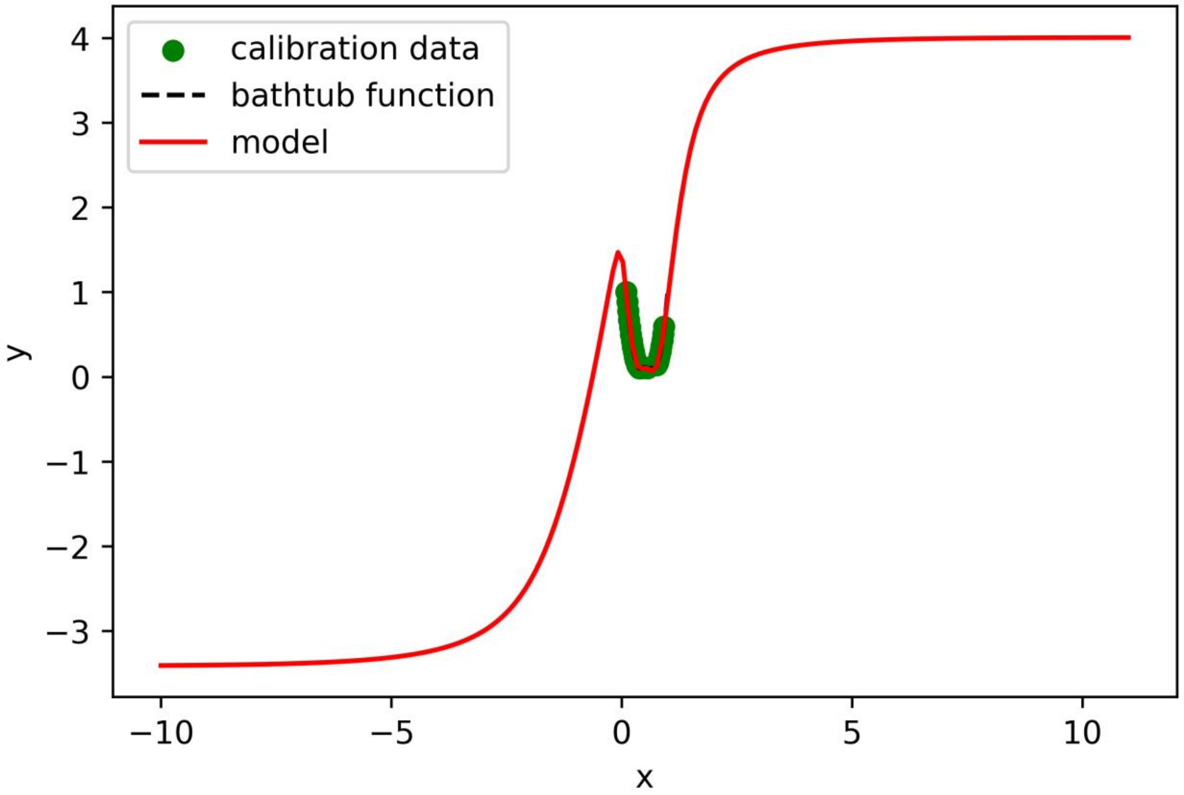


- Infinitely differentiable



ACTIVATION FUNCTIONS - TANH

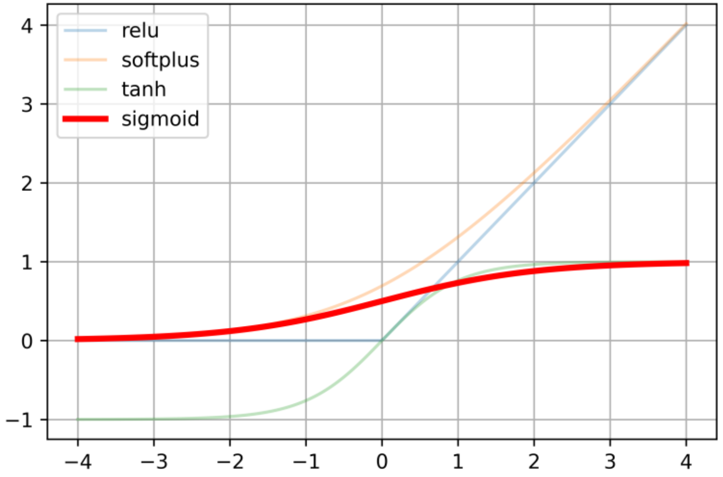


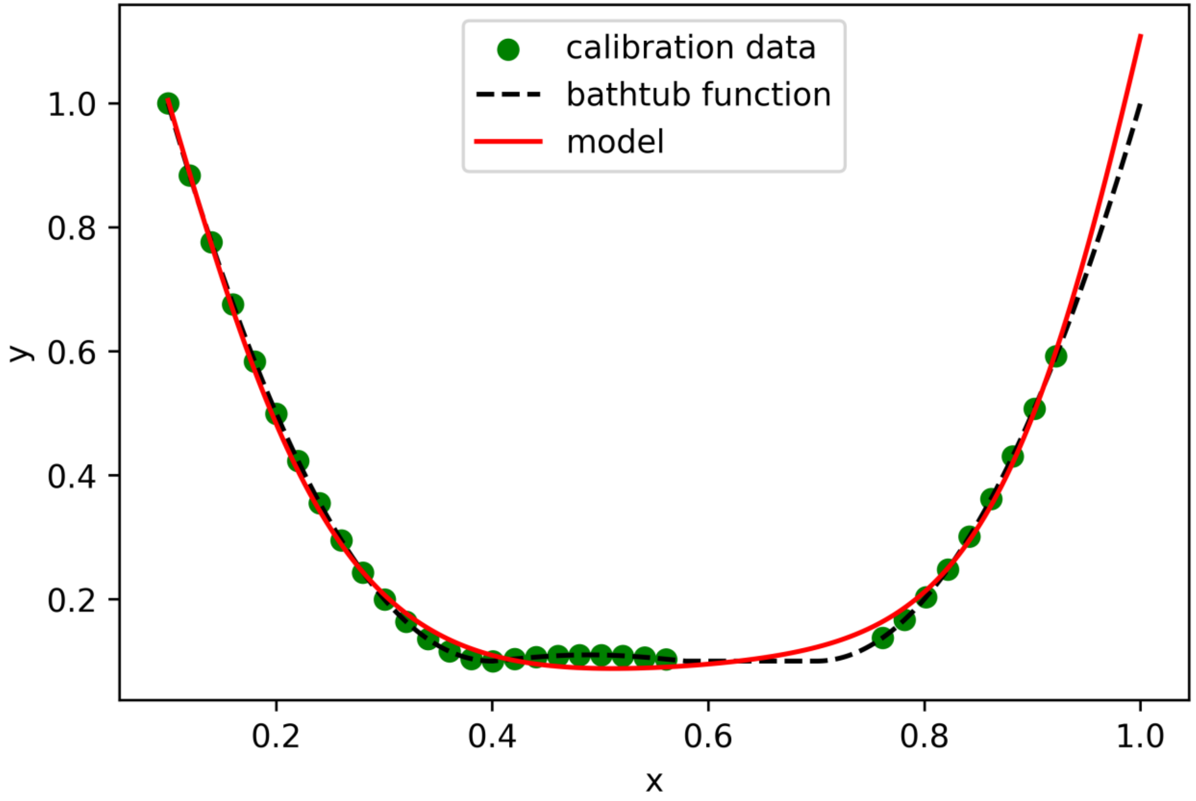


- Infinitely differentiable
- Bounded

HYPERPARAMETER SWEEP

ACTIVATION FUNCTIONS - SIGMOID

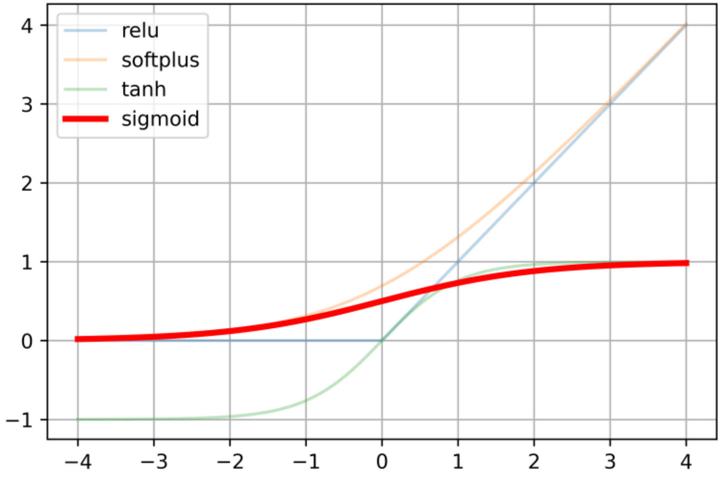




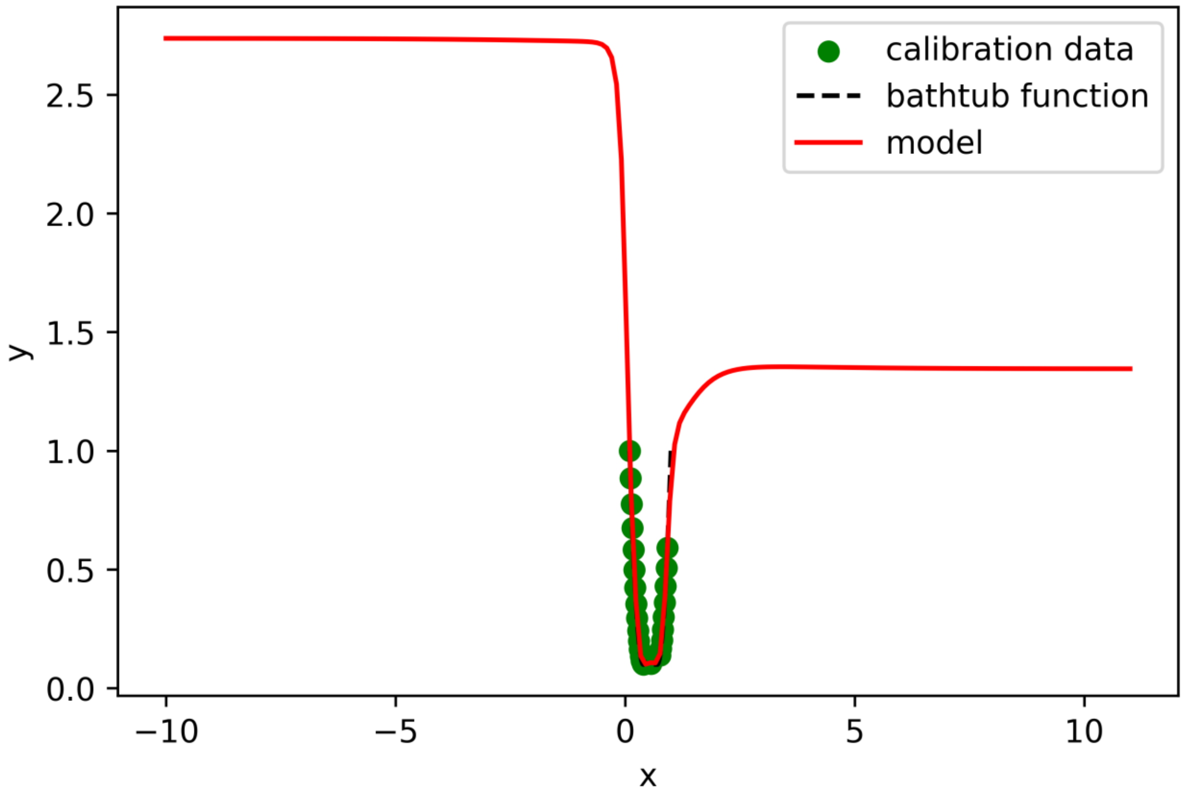
- Infinitely differentiable

HYPERPARAMETER SWEEP

ACTIVATION FUNCTIONS - SIGMOID

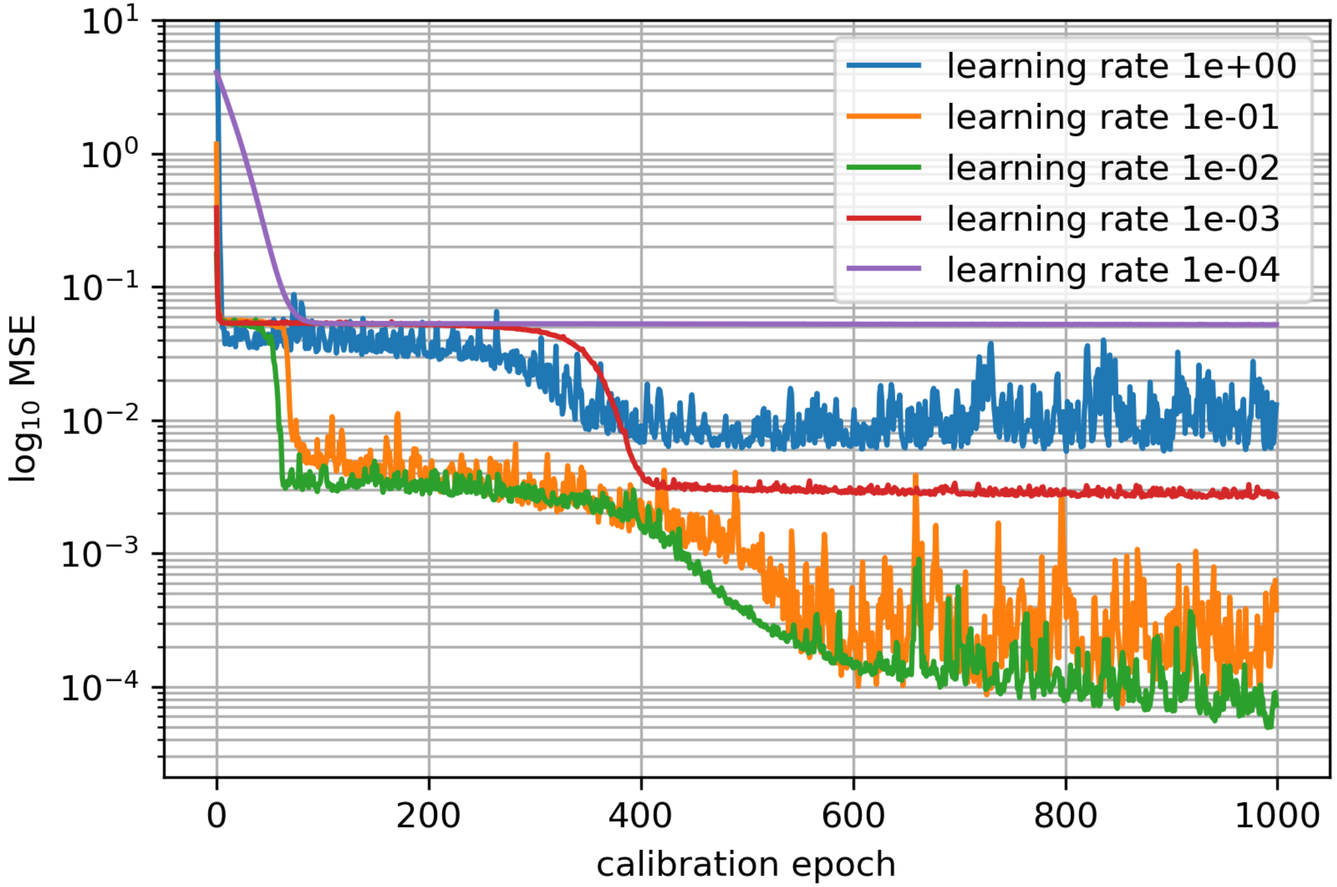


- Infinitely differentiable
- Bounded



HYPERPARAMETER SWEEP

LEARNING RATE



- Problem specific optimal learning rate
- Larger models require smaller learning rates

CONCLUSION

- **Model Complexity:** Complex “enough”
- **Number of epochs:** “Enough” to converge
- **Activation function:** Depends on use case. Differentiability, boundedness, convexity, …

All parameters affect each other.
Experimentation/parameter studies necessary

TECHNISCHE
UNIVERSITÄT
DARMSTADT

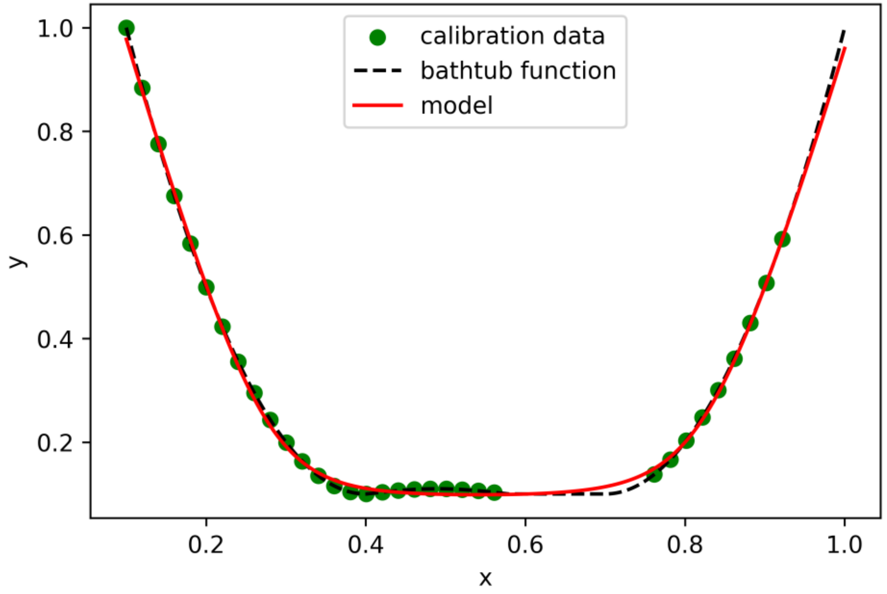
TASK 1 – 1.2

INPUT CONVEX NEURAL NETWORKS

INPUT CONVEX NEURAL NETWORKS

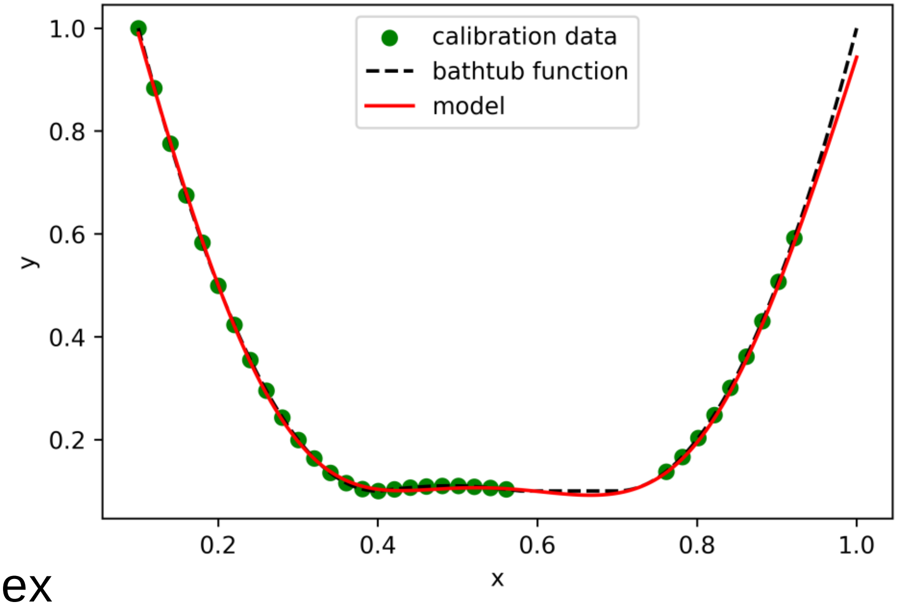
COMPARISON

ICNN



- Convex
- Slower convergence
- Up to 50% zero-weights
- Physically sensible inter- and extrapolation

FFNN

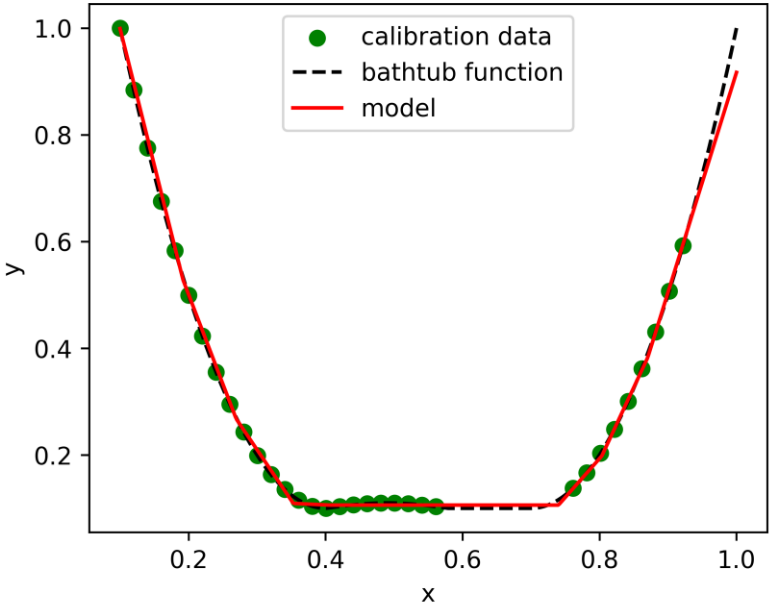


- Not convex
- Faster convergence
- Usually no zero-weights
- Unpredictable inter- and extrapolation behavior

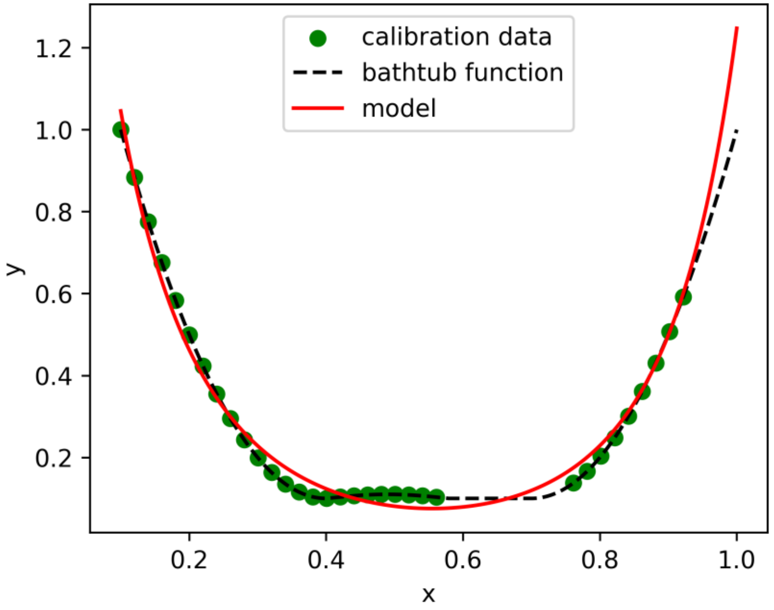
INPUT CONVEX NEURAL NETWORKS

ALTERNATIVE ICNN ACTIVATION FUNCTIONS

All f such that $\frac{d^2 f}{dx^2} > 0, \frac{df}{dx} > 0$. E.g.,



Relu $f(\boldsymbol{x}) = \max(0, \boldsymbol{x})$



Exp $f(\boldsymbol{x}) = e^{\boldsymbol{x}}$

● ● ●

TASK 1 – 2.2

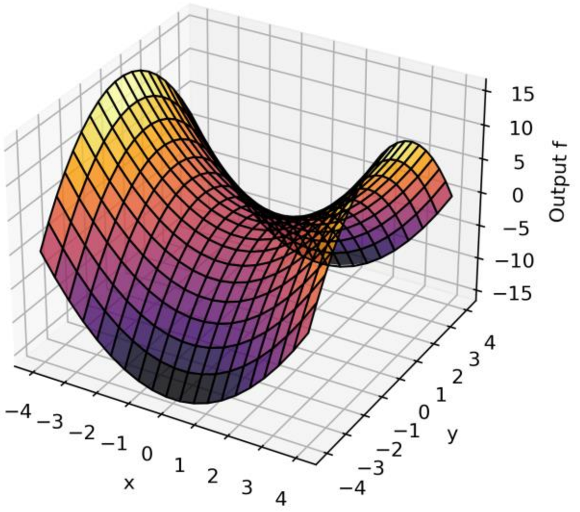
TRAINABLE CUSTOM LAYER

TRAINABLE CUSTOM LAYER

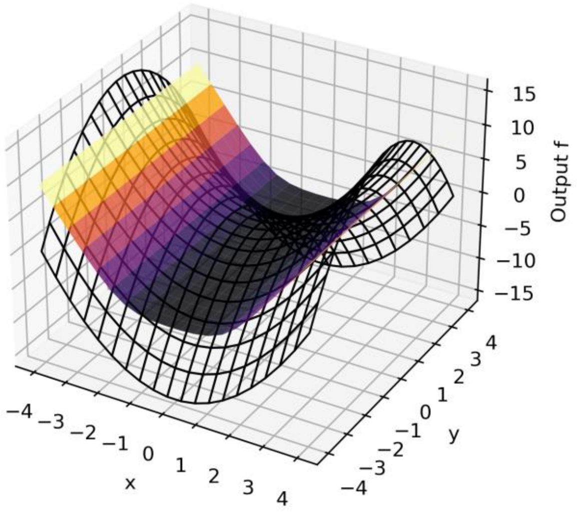
COMPARISON

NON-CONVEX DATA

FFNN

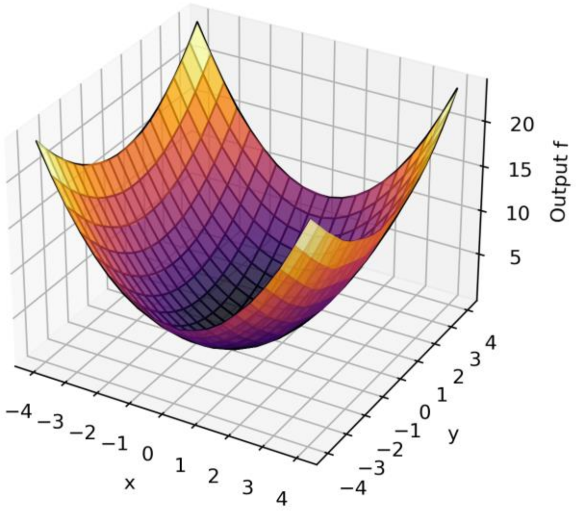


ICNN

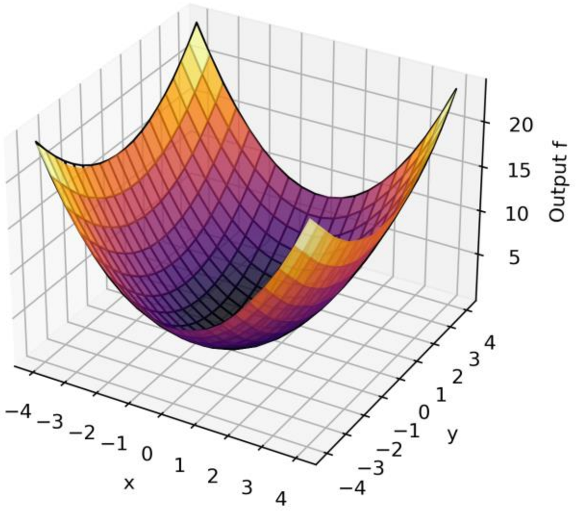


CONVEX DATA

FFNN



ICNN



TASK 1 - 2.3

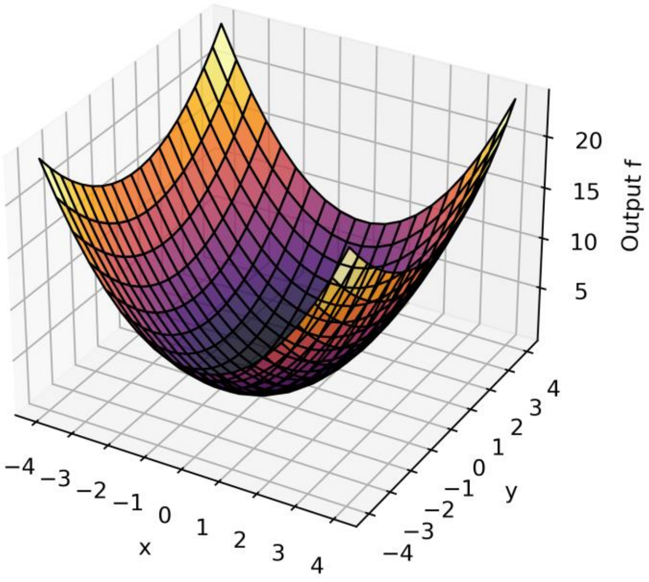
SOBOLEV TRAINING

SOBOLEV TRAINING

PREDICTION

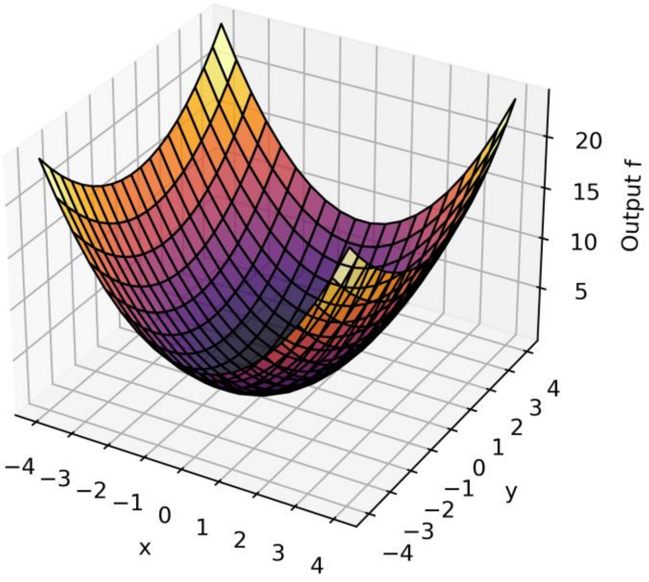
FUNCTION ONLY

Loss 1.343e-02



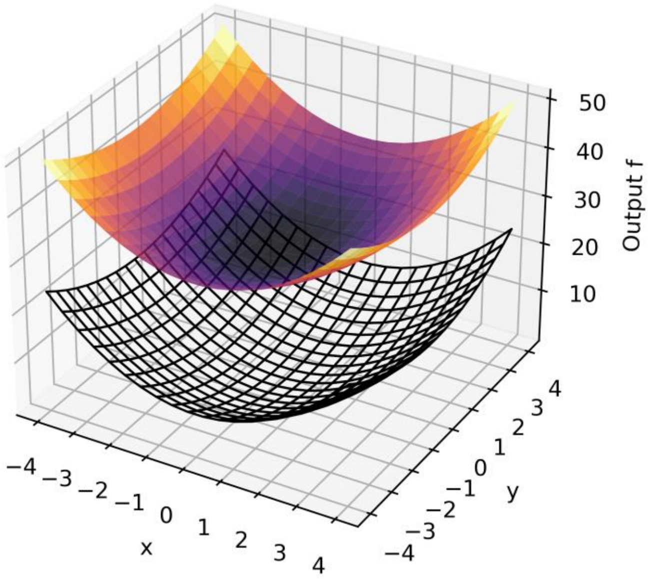
FUNCTION AND GRADIENT

Loss 1.437e-03



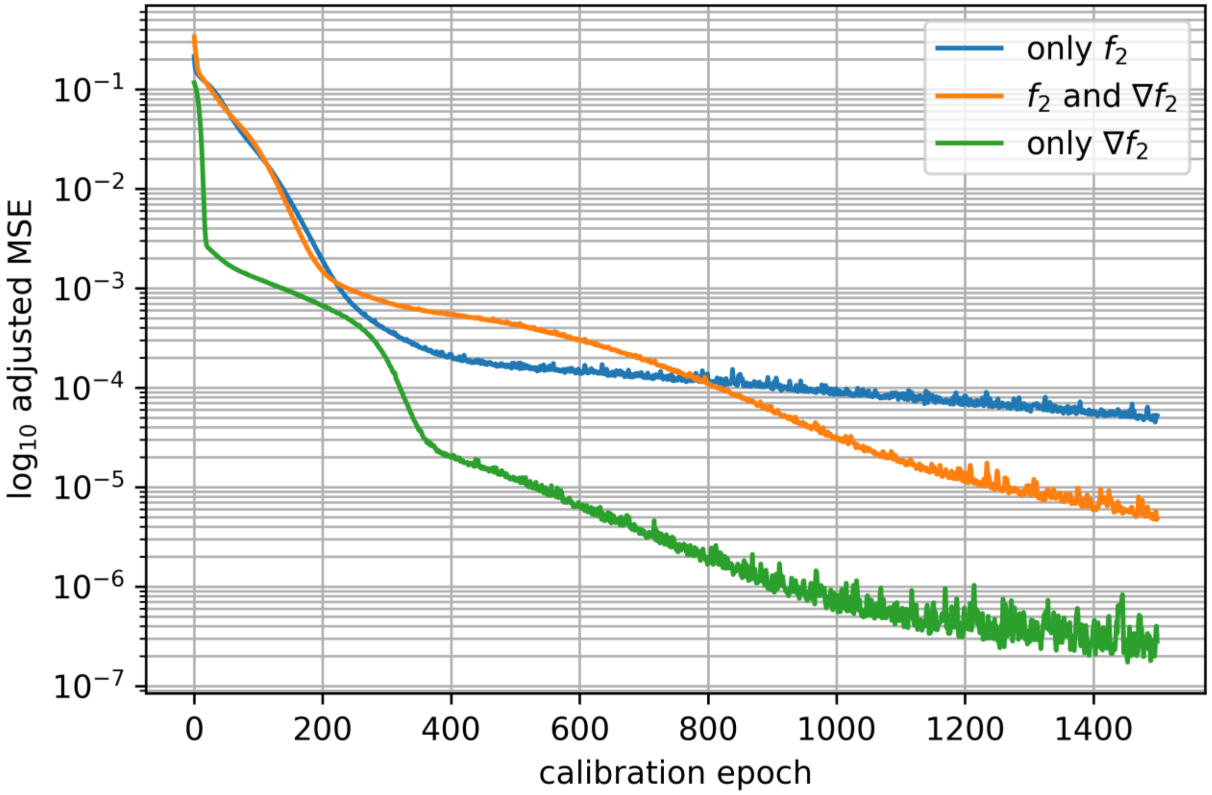
GRADIENT ONLY

Loss 1.045e-04

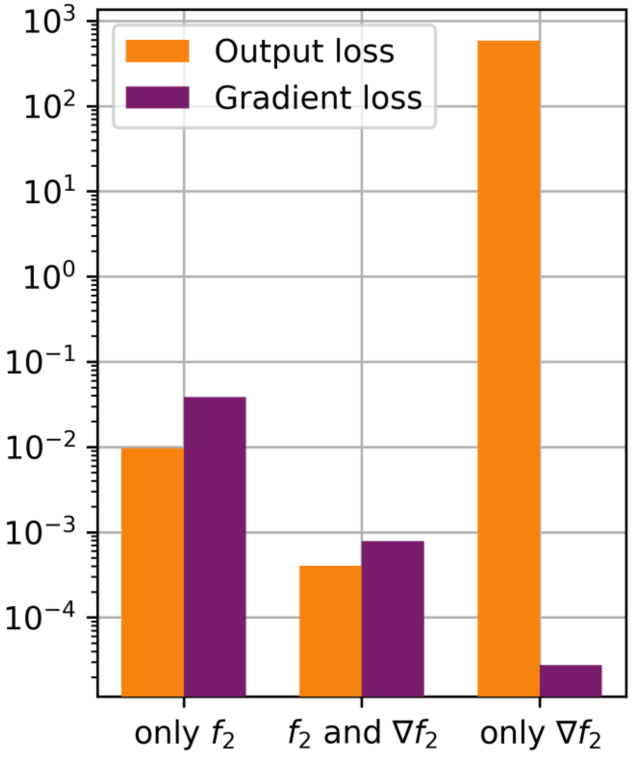


LOSS COMPARISON

TRAINING LOSS



EVALUATION LOSS



DISCUSSION