Machine Learning Surrogates with OMLT and IDAES for Improved Design and Analysis of Energy Systems

Bashar L. Ammari,[1] Francesco Ceccon,[2] Jordan Jalving,[3] Joshua Haddad,[3] Alexander Thebelt,[2] Calvin Tsay,[2] Michael L. Bynum,[3] Carl D. Laird,[1] Ruth Misener[2]

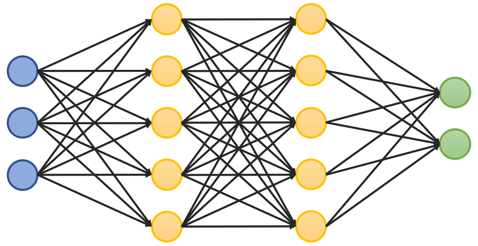
[1] Carnegie Mellon University [2] Imperial College London [3] Sandia National Laboratories

October 17, 2022

Machine Learning (ML) Surrogates and Optimization

Dose	Weight	Age	Effectiveness
…	…	…	…

PyTorch scikit learn TensorFlow



$$\begin{aligned} \min \quad & f(x,y,z,\theta) \\ \text{s.t.} \quad & g(x,y,z,\theta) \leq 0 \\ & h(x,y,z,\theta) = 0 \\ & z = ML(\theta) \end{aligned}$$

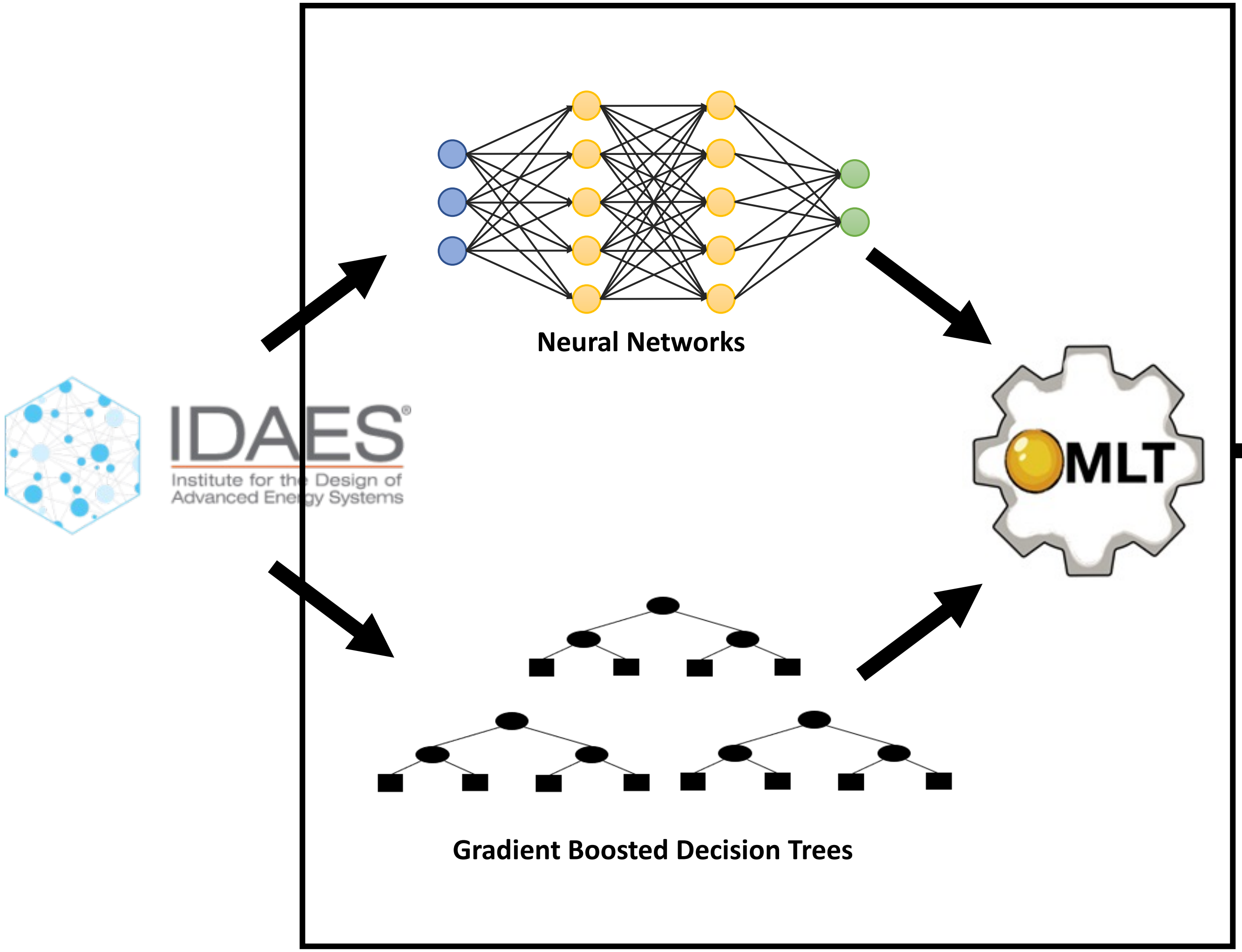
Mixed-Integer Linear and Nonlinear Programming

?

Motivation

- ML surrogates are powerful in their ability to describe complex functional relationships
- New optimization formulations allow decision-making with trained ML models e.g.
  - Verification of neural network surrogates (Haddad et al., 2022)
  - Sequential auction optimization (Verwer et al., 2017)
  - Personalized chemotherapy regimens (Bertsimas et al., 2016)
- Embedding ML surrogates into optimization problems allows for utilization of global optimization techniques.

OMLT: Optimization and Machine Learning Toolkit



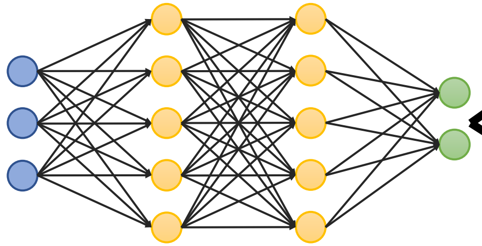
1) Overview of OMLT and its capabilities.

2) We will show how IDAES and OMLT can integrate on either end of the workflow

- Autothermal Reformer Example
- Optimal Heat Exchanger Design

3) Discussion of other potential use cases for OMLT e.g. flexibility analysis

Neural Network Formulations



Smooth Activation ⇒ NLP

ReLU Activation ⇒ MIP

Smooth Activations: Full Space vs Reduced Space

For a network with m inputs, r outputs, and $N-m$ hidden nodes,

x : inputs
y : outputs
$\sigma(\cdot)$: activation fn
$z, \hat{z}$: intermediates

$$x_i = z_i \qquad \forall i \in \{1, ..., m\}$$

$$\hat{z}_i = \sum_{j=1}^{i-1} w_{ij} z_j + b_i \qquad \forall i \in \{m+1, ..., N\}$$

$$z_i = \sigma(\hat{z}_i) \qquad \forall i \in \{m+1, ..., N\}$$

$$y_i = \sum_{j=1}^{N} w_{ij} z_j + b_i \qquad \forall i \in \{N+1, ..., N+r\}$$

- **Full space** formulation includes all the intermediate equalities and variables
  - Results in a large-scale optimization problem (branch-and-bound)
- **Reduced space** formulation generates functional form and eliminates intermediate variables.
  - Can be difficult to generate large expression tree in the algebraic modelling language
  - Often have weaker relaxations when propagated through large neural networks (Schweidtmann & Mitsos, 2019)

ReLU Mixed-Integer Linear Programming Formulations

MIP representation based on Big-M (Fischetti and Jo, 2018)

Extensions:

- Strong reformulations – convex hull (Anderson et al., 2020)
- Partition-based formulations (Tsay et al., 2021)

Example for a single neuron

$$\begin{bmatrix} \sigma \\ y = 0 \\ w^T x + b \leq 0 \end{bmatrix} \bigvee \begin{bmatrix} \neg\sigma \\ y = w^T x + b \\ w^T x + b \geq 0 \end{bmatrix}$$

Big M

$$y \geq (w^T x + b)$$
$$y \leq (w^T x + b) - (1 - \sigma) LB^0$$
$$y \leq \sigma UB^0$$

Partition-based

$$\begin{bmatrix} \sigma \\ y = 0 \\ \sum_{p=1}^{P} z_p + b \leq 0 \end{bmatrix} \bigvee \begin{bmatrix} \neg\sigma \\ y = \sum_{p=1}^{P} z_p + b \\ y \leq 0 \end{bmatrix}$$

[Tjandraatmadja, Christian, et al. "The convex relaxation barrier, revisited: Tightened single-neuron relaxations for neural network verification." Advances in Neural Information Processing Systems 33 (2020): 21675-21686.]

Gradient Boosted Decision Trees

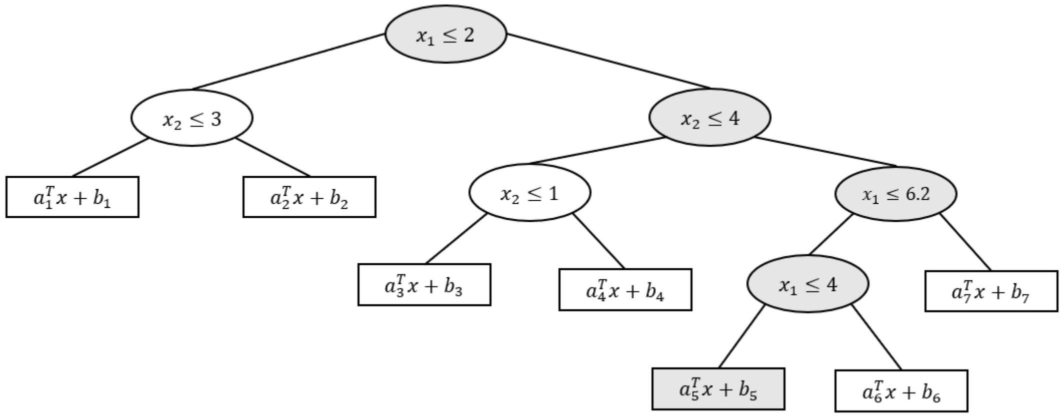
Mixed-Integer Linear Program (MILP)

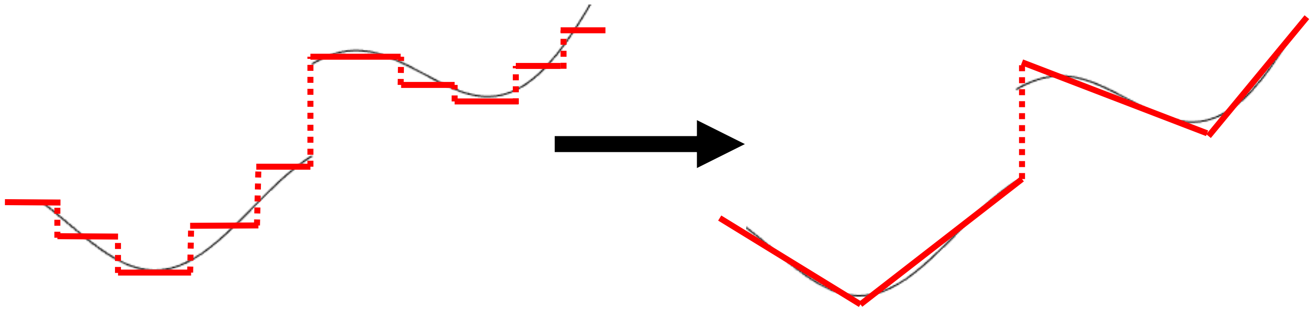
min	$\sum_{t\in T}\sum_{\ell\in L} F_{t,\ell} z_{t,\ell}$		Return the value of the selected leaf $F_{t,\ell}$ for each tree
s.t.	$\sum_{\ell\in L} z_{t,\ell} = 1$	$\forall t \in T$	Only select one leaf per tree in the ensemble
	$\sum_{\ell\in \text{Left}_s} z_{t,\ell} \leq y_{i(s),j(s)}$	$\forall t \in T, \quad s \in S$	Select a leaf only if all the corresponding splits occur
	$\sum_{\ell\in \text{Right}_s} z_{t,\ell} \leq 1 - y_{i(s),j(s)}$	$\forall t \in T, \quad s \in S$	
	$y_{i,j} \leq y_{i,j+1}$	$\forall i \in [n], \quad j \in [m_i - 1]$	Ensure splits occur in the correct order
	$x_i \geq v_{i,0} + \sum_{j=1}^{m_i} (v_{i,j} - v_{i,j-1})(1 - y_{i,j})$	$\forall i \in [n]$	Link x_i to the correct interval defined by the splits corresponding to the selected leaf
	$x_i \leq v_{i,m_i+1} + \sum_{j=1}^{m_i} (v_{i,j} - v_{i,j+1}) y_{i,j}$	$\forall i \in [n]$	
	$x \in \mathbb{R}^n$		
	$y_{i,j} \in \{0,1\}$	$\forall i \in [n], j \in [m_i]$	
	$z_{t,\ell} \in \{0,1\}$	$\forall t \in T, \quad \ell \in L$	

[Mistry et al. 2021], [Mišić, 2020]

Support for Linear Model Decision Trees Coming Soon!

Linear Model Decision Trees





Formulations

$$\bigvee \begin{bmatrix} Z_\ell \\ x_i \le v^U_{i,\ell} \quad \forall\, i \in [n] \\ x_i \ge v^L_{i,\ell} \quad \forall\, i \in [n] \\ a_\ell^T x + b_\ell = d \end{bmatrix}$$

$$\begin{aligned} \text{s.t.} \quad & d = \sum_{\ell \in L} (a_\ell^T x + b_\ell) z_\ell \end{aligned} \quad \Longleftrightarrow \quad \begin{aligned} \text{s.t.} \quad & a_\ell^T x + b_\ell \le d + M_\ell^U (1 - z_\ell) && \forall \ell \in L \\ & a_\ell^T x + b_\ell \ge d - M_\ell^L (1 - z_\ell) && \forall \ell \in L \end{aligned}$$

$$\begin{aligned}
& \sum_{\ell \in L} z_\ell = 1 \\
& \sum_{\ell \in \text{Left}_s} z_\ell \le y_{i(s),j(s)} && \forall s \in S \\
& \sum_{\ell \in \text{Right}_s} z_\ell \le 1 - y_{i(s),j(s)} && \forall s \in S \\
& y_{i,j} \le y_{i,j+1} && \forall i \in [n], j \in [m_i - 1] \\
& x_i \ge v_{i,0} + \sum_{j=1}^{m_i} (v_{i,j} - v_{i,j-1})(1 - y_{i,j}) && \forall i \in [n] \\
& x_i \le v_{i,m_i+1} + \sum_{j=1}^{m_i} (v_{i,j} - v_{i,j+1}) y_{i,j} && \forall i \in [n] \\
& x \in \mathbb{R}^n \\
& y_{i,j} \in \{0,1\} && \forall i \in [n], j \in [m_i] \\
& z_\ell \in \{0,1\} && \forall \ell \in L
\end{aligned}$$

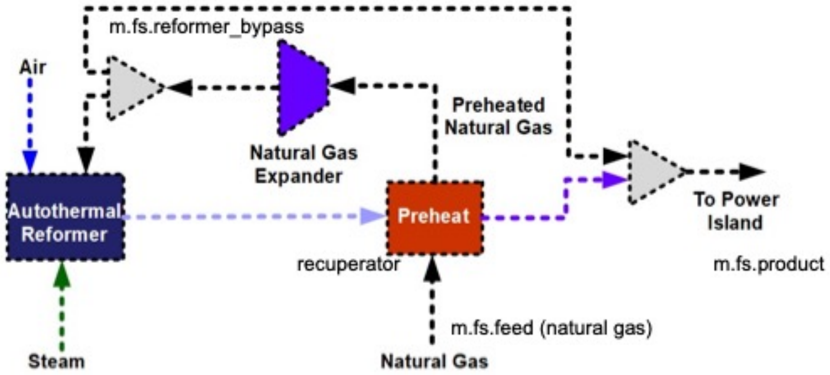
Optimization of an Autothermal Reformer

The Idea

Replace complex first principles models with data driven surrogates.

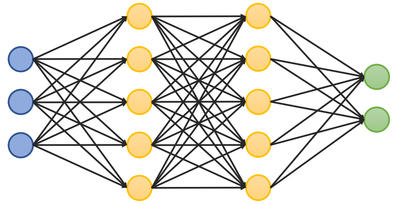
Motivation

1) Some first principles models are calculated through compiled codes (thermodynamics) and replacing them with surrogates makes the problem completely equation oriented.
2) High fidelity models may also have difficult non-convex and non-linear properties



Data Generated using the IDAES Framework

Inputs: Reformer bypass fraction, ratio of natural gas (NG) to steam
Outputs: Steam flow, reformer duty, [Ar], [C_2H_6], [C_3H_8], [C_4H_{10}], [CH_4], [CO], [CO_2], [H_2], [H_2O], [N_2]



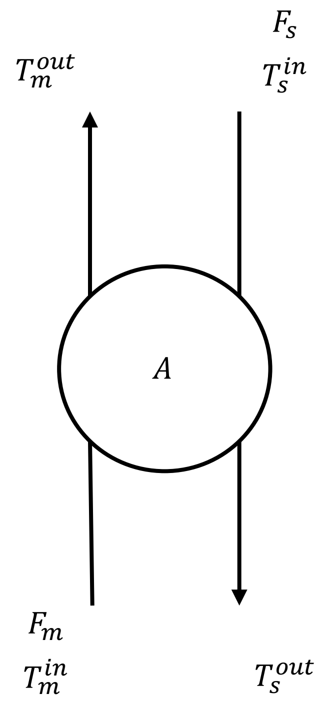
OMLT

$$\max \quad [H_2] \qquad \text{s.t.} \quad [N_2] \leq 0.34$$

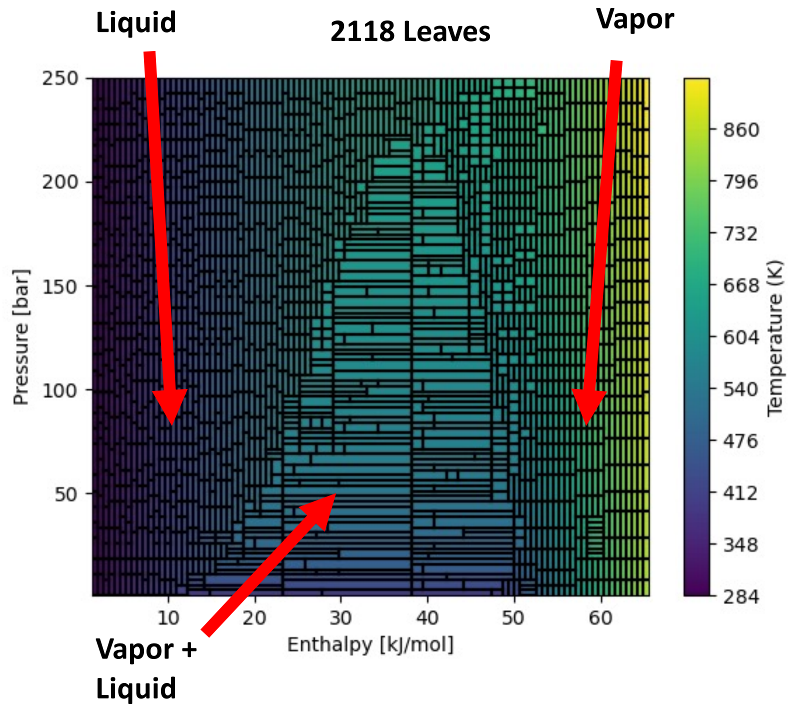
	Sigmoid	ReLU
Bypass Fraction	0.1	0.1
NG/Steam Ratio	1.12	1.12
[H_2]	0.331	0.331
[N_2]	0.34	0.34

IDAES
Institute for the Design of Advanced Energy Systems

Optimal Design of a Heat Exchanger



$$
\begin{aligned}
\min \quad & K_s F_s + K_A A \\
\text{s.t.} \quad & Q = UA\Delta T_{LM} \\
& \Delta T_{LM} = (\Delta T_1 \Delta T_2 \frac{\Delta T_1 + \Delta T_2}{2})^{\frac{1}{3}} \\
& \Delta T_1 = T_s^{out} - T_m^{in} \\
& \Delta T_2 = T_s^{in} - T_m^{out} \\
& Q = F_m C_p (T_m^{out} - T_m^{in}) \\
& Q = -F_s (H_s^{out} - H_s^{in}) \\
& T_s^{in} = f(P_s^{in}, H_s^{in}) \\
& T_s^{out} = f(P_s^{out}, H_s^{out})
\end{aligned}
$$



Motivation

- IDAES currently performs thermodynamic calculations using external functions and compiled C++ codes which prevents use of global solvers such as BARON

The Idea

- Train an ML surrogate on steam properties and embed the model into a formulation capable of **global optimization** to compare with **locally optimal IDAES model**.

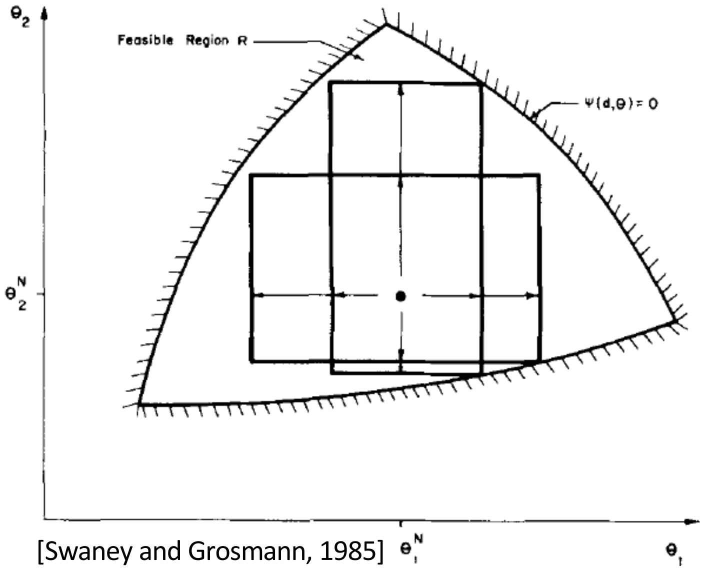
	Solver	A $[m^2]$	F_s $[\frac{kg}{s}]$	Annual Cost $[\frac{\$}{yr}]$
IDAES Model	IPOPT	2924	60.31	4.78 million
Linear Tree	BARON	2926	60.27	4.78 million

Applications to Flexibility Analysis

Feasibility Test
Given a design d and uncertain parameters θ, is there a control action z I can take to ensure all constraints $g_j(d, z, \theta) \leq 0$ are satisfied within expected uncertainty $\theta^L \leq \theta \leq \theta^U$.

$$\chi(d) = \max_{\theta} \min_{z} \quad g(d, z, \theta)$$

$$\begin{aligned} \chi(d) = \max_{\theta} \quad & g(d, z, \theta) \\ \text{s.t.} \quad & z = ML(\theta) \end{aligned}$$



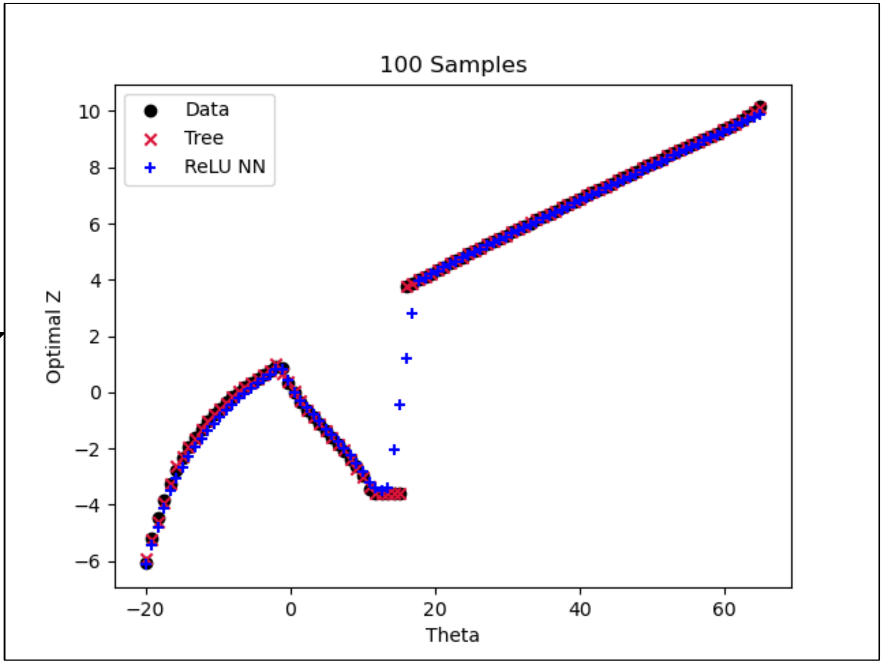
[Swaney and Grosmann, 1985]

Current Solution Strategies
- If constraints meet the sufficient conditions for vertex solutions, can use vertex enumeration (Swaney and Grossmann, 1985)
- If convex, active set strategy may improve scalability (Grossmann & Floudas, 1986)
- If non-convex, must use global branch and bound approach for the active set strategy (Floudas et al., 2001)
- If mixed-integer linear or mixed-integer convex quadratic, a multi-parametric optimization approach proposed by (Avraamidou & Pistikopoulos, 2019)

The Idea
Use Machine Learning to learn the inner control problem and convert to a single level optimization problem
Motivation
Potential to **improve scalability** and **can provide a guaranteed conservative estimate**

Applications to Flexibility Analysis

$$\begin{aligned}
\min \quad & \delta \\
\text{s.t.} \quad & 0.01(z-1.5)^4 - 0.05(z-1.5)^3 - (z-1.5)^2 - (z-1.5) - 10 + \theta = s_1 \\
& (-0.02\theta - 14)z + (1.66\theta - 100) = s_2 \\
& 30z - 50 - 4\theta + e^{(-0.2\theta+1)} = s_3 \\
& -10 - z = s_4 \\
& z - 15 = s_5 \\
& \sum_{j \in J} s_j y_j = 0 \\
& \sum_{j \in J} y_j \geq 1 \\
& \boxed{z = ML(\theta)} \\
& \theta^N - \delta\Delta\theta^- \leq \theta \leq \theta^N + \delta\Delta\theta^+ \\
& -20 \leq \theta \leq 65 \\
& y_j \in \{0,1\}, \quad s_j \leq 0 \qquad \forall j
\end{aligned}$$



$$\min_z g(d, z, \theta)$$

$\theta^N = 22.5$
$\Delta\theta^+ = 42.5$
$\Delta\theta^- = -42.5$

Method	δ
True Solution	1
Vertex	1
ReLU NN	0.152
Linear Tree	0.890
Active Set	0.300

Conclusions

- Since the surrogate is at best the exact control policy, the calculated index is conservative
  - Desirable property for safety considerations in process design
- Linear model decision tree gives a better solution than ReLU NN because it can model discontinuities

Conclusions and Key Ideas

Key Ideas

- ML Surrogates can model complex behavior inherent within data
- We can represent ML surrogates using mathematical programming (neural networks, tree ensembles etc…)
- OMLT streamlines embedding ML surrogates in optimization problems
- IDAES and OMLT can integrate to extract data (HX Design) or even substitute for a unit in a flow sheet (Autothermal Reformer)
- Using ML surrogates to model the optimal control policy in flexibility analysis can provide a conservative estimate

Acknowledgements

Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Sandia National Laboratories

IDAES® Institute for the Design of Advanced Energy Systems

References

[1] Anderson, R., Huchette, J., Ma, W., Tjandraatmadja, C., and Vielma, J. P. (2020). Strong mixed-integer programming formulations for trained neural networks. Mathematical Programming, 183:3–39.

[2] Avraamidou, S. and Pistikopoulos, E. N. (2019). A multi-parametric optimization approach for bilevel mixed-integer linear and quadratic programming problems. Computers Chemical Engineering, 125:98–113.

[3]Bertsimas, D., O'Hair, A., Relyea, S., and Silberholz, J. (2016). An analytics approach to designing combination chemotherapy regimens for cancer. Management Science, 62:1511–1531.

[4] Ceccon, F., Jalving, J., Haddad, J., Thebelt, A., Tsay, C., Laird, C. D., and Misener, R. (2022). OMLT: Optimization machine learning toolkit.

[5] Fischetti, M. and Jo, J. (2018). Deep neural networks and mixed integer linear optimization. Constraints, 23:296–309.

[6] Floudas, C. A., G¨um¨us¸ Z. H., and Ierapetritou, M. G. (2001). Global optimization in design under uncertainty: Feasibility test and flexibility index problems. Industrial and Engineering Chemistry Research, 40:4267–4282.

[7] Grossmann, I. E. and Floudas, C. A. (1987). Active constraint strategy for flexibility analysis in chemical processes. Comput. Chem. Eng, II:675–693.

[8] Haddad, J., Bynum, M., Eydenberg, M., Blakely, L., Kilwein, Z., Boukouvala, F., Laird, C. D., and Jalving, J. (2022). Verification of neural network surrogates. Computer Aided Chemical Engineering, 51:583–588.1

[9] Mistry, M., Letsios, D., Krennrich, G., Lee, R. M., and Misener, R. (2021). Mixed-integer convex nonlinear optimization with gradient-boosted trees embedded. INFORMS Journal on Computing, 33:1103–1119.

[10] Misic, V. V. (2020). Optimization of tree ensembles. Operations Research, 68:1605–1624.

[11] Schweidtmann, A. M. and Mitsos, A. (2019). Deterministic global optimization with artificial neural networks embedded. Journal of Optimization Theory and Applications, 180:925–948.

[12] Swaney, R. E. and Grossmann, I. E. (1985). An index for operational flexibility in chemical process design. AIChE Journal, 31:621–630.

[13] Tsay, C., Kronqvist, J., Thebelt, A., and Misener, R. (2021). Partition-based formulations for mixed-integer optimization of trained ReLU neural networks.

[14] Verwer, S., Zhang, Y., and Ye, Q. C. (2017). Auction optimization using regression trees and linear models as integer programs. Artificial Intelligence, 244:368–395.