Aalto-yliopisto
Perustieteiden korkeakoulu

Reformulating neural networks as mathematical programming problems

Joonatan Linkola

17.4.2022

Advisor: Nikita Belyak

Supervisor: Fabricio Oliveira

Systeemianalyysin laboratorio

Background

- Some processes are difficult to optimize

Background

- Some processes are difficult to optimize
- Machine learning (ML) models as surrogates

Background

- Some processes are difficult to optimize
- Machine learning (ML) models as surrogates
- Neural networks (NN) can be represented as a set of linear constraints and binary variables – Mixed integer linear program (MILP)

Background

- Some processes are difficult to optimize
- Machine learning (ML) models as surrogates
- Neural networks (NN) can be represented as a set of linear constraints and binary variables – Mixed integer linear program (MILP)

$$\begin{aligned}
\textbf{Maximize} \quad & x_1 + x_2 + z && \text{(Objective function)} \\
\text{s.t. } \quad & 2x_1 + 4x_2 \leq 8 && \\
& 3x_1 + 5x_2 + z \leq 12 && \text{(Constraints)} \\
& x_1, x_2 \geq 0. && \\
& z \in \mathbb{Z}^+ && \text{(Variables)}
\end{aligned}$$

Background

- Some processes are difficult to optimize
- Machine learning (ML) models as surrogates
- Neural networks (NN) can be represented as a set of linear constraints and binary variables – Mixed integer linear program (MILP)
- For a given input, the surrogate is mapped to a feasible solution of constraints, giving the same output as the NN

Background

- Some processes are difficult to optimize
- Machine learning (ML) models as surrogates
- Neural networks (NN) can be represented as a set of linear constraints and binary variables – Mixed integer linear program (MILP)
- For a given input, the surrogate is mapped to a feasible solution of constraints, giving the same output as the NN
  - Optimize the input to maximize the output

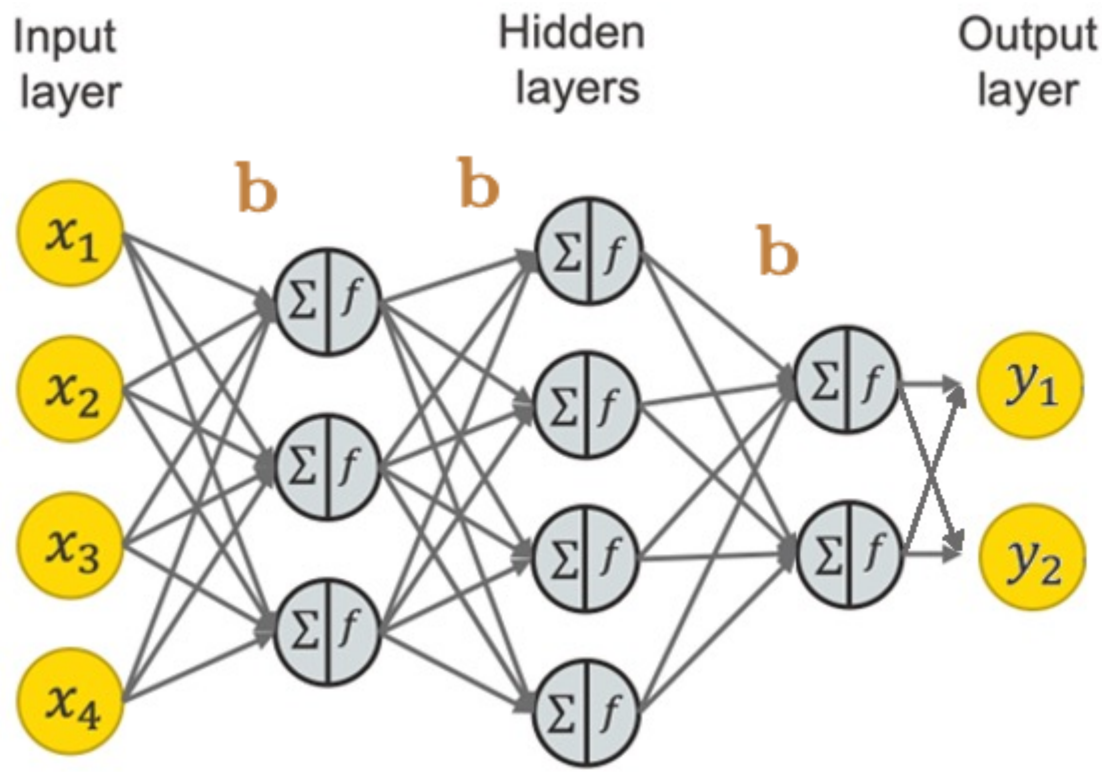
Background

- Some processes are difficult to optimize
- Machine learning (ML) models as surrogates
- Neural networks (NN) can be represented as a set of linear constraints and binary variables – Mixed integer linear program (MILP)
- For a given input, the surrogate is mapped to a feasible solution of constraints, giving the same output as the NN
    - Optimize the input to maximize the output
    - Add additional constraints

Background

- Some processes are difficult to optimize
- Machine learning (ML) models as surrogates
- Neural networks (NN) can be represented as a set of linear constraints and binary variables – Mixed integer linear program (MILP)
- For a given input, the surrogate is mapped to a feasible solution of constraints, giving the same output as the NN
  - Optimize the input to maximize the output
  - Add additional constraints
  - etc.

Neural network (ReLU network) structure



$$f(x) = \mathbf{ReLU}(x) := \max\{0, x\} \text{ (componentwise)}$$

MILP formulation for ReLU networks (Grimstad and Andersson, 2019)

Input layer:

$$L_{\text{in}} \leq x^0 \leq U_{\text{in}}$$

Hidden ReLU layers:

$$W^k x^{k-1} + b^k = x^k - s^k, \quad x^k, s^k \geq 0$$

$$x^k = 0 \;\vee\; s^k = 0$$

Output layer:

$$W^K x^{K-1} + b^K = x^K$$

$$L_{\text{out}} \leq x^K \leq U_{\text{out}}$$

W: weights at each layer
b: biases at each layer
x: node value at each layer
$k = \{0, \ldots, K\}$: node index

Aims

- Develop a model which represents a trained ReLU NN as a MILP problem in Julia

Aims

- Develop a model which represents a trained ReLU NN as a MILP problem in Julia
- Test the model against NNs of non-linear functions and digit image classification problems

Aims

- Develop a model which represents a trained ReLU NN as a MILP problem in Julia
- Test the model against NNs of non-linear functions and digit image classification problems
- Use the model to build adversarial digit images to enhance the original NN

Remarks

- Only ReLU networks are considered
  - Convolutional layers and other activation functions cannot be represented with the same MILP formulation

Remarks

- Only ReLU networks are considered
  - Convolutional layers and other activation functions cannot be represented with the same MILP formulation

- Model must be computationally feasible
  - Size of the model and its run time should be reasonable

Tools

- Julia, namely Flux and JuMP libraries
- MNIST handwritten digit database

Sources

- ReLU networks as surrogate models in mixed-integer linear programs (Grimstad and Andersson, 2019)
- Deep neural networks and mixed integer linear optimization (Fischetti and Jo, 2018)
- MNIST handwritten digits database

Schedule

- 11/2022: Studying the topic and related papers
- 2/2023: Start implementing the MILP formulation in Julia
- 4/2023: Presentation of the thesis topic
- 6/2023: Results and thesis ready