Aalto-yliopisto
Perustieteiden
korkeakoulu

# P-split formulation for neural networks
## (topic presentation)

*Venla Kjäll*

*20.3.2024*

Advisor: *Yu Liu*
Supervisor: *Fabricio Oliveira*



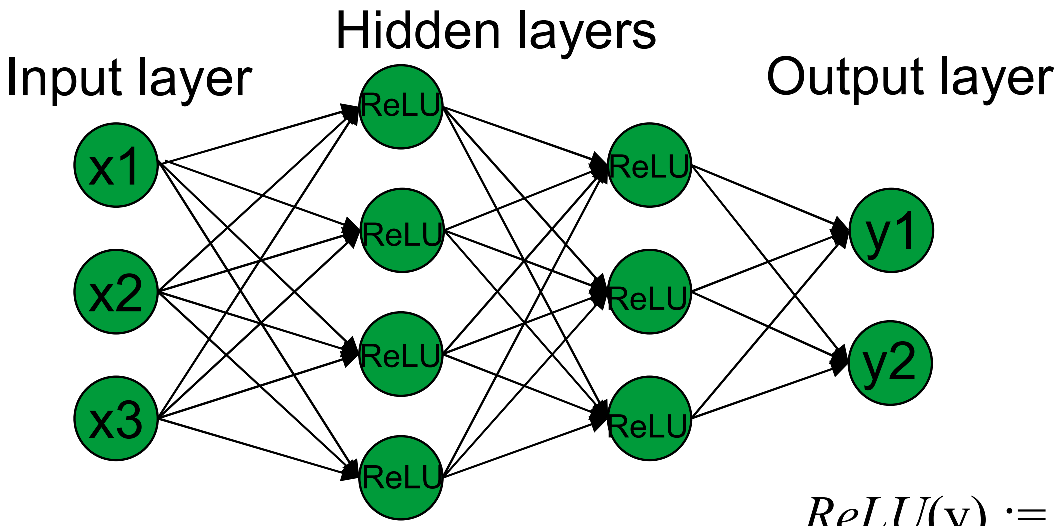
Systeemianalyysin
laboratorio

# Table of contents


- Neural network, ReLU
- 0-1 MILP for neural networks
- P-split formulation
- Objective
- Tools
- Schedule
- References and literature

# Background – Neural network structure (ReLU)

- The hidden layers use only ReLU activation functions
- Output layer uses the identity activation
- ReLU function can be modelled using indicator constraints (or big-M constraints)




$$ReLU(y) := \max\{0, y\} \text{ (componentwise)}$$

# Background – 0-1 MILP formulation

- Not suitable for training, it is useful for finding optimal input examples for a given trained neural network

$$\min \sum_{k=0}^{K} \sum_{j=1}^{n_k} c_j^k x_j^k + \sum_{k=1}^{K} \sum_{j=1}^{n_k} \gamma_j^k z_j^k$$

$$\left.\begin{aligned}
\sum_{i=1}^{n_{k-1}} w_{ij}^{k-1} x_i^{k-1} + b_j^{k-1} &= x_j^k - s_j^k \\
x_j^k, s_j^k &\geq 0 \\
z_j^k &\in \{0, 1\} \\
z_j^k = 1 \rightarrow x_j^k &\leq 0 \\
z_j^k = 0 \rightarrow s_j^k &\leq 0
\end{aligned}\right\} k = 1, \ldots, K, \;\; j = 1, \ldots, n_k$$

$$lb_j^0 \leq x_j^0 \leq ub_j^0, \qquad j = 1, \ldots, n_0$$

$$\left.\begin{aligned}
lb_j^k \leq x_j^k \leq ub_j^k \\
\overline{lb}_j^k \leq s_j^k \leq \overline{ub}_j^k
\end{aligned}\right\} k = 1, \ldots, K, \;\; j = 1, \ldots, n_k.$$

Matteo Fischetti, Jason Jo, (2018). Deep neural networks and mixed integer linear optimization

# Backgroud – P-split formulation

- ReLU activation function can also be represented as a disjunction: $\begin{bmatrix} y=w^T x+b \\ w^T x+b \geq 0 \end{bmatrix} \vee \begin{bmatrix} y=0 \\ w^T x+b \leq 0 \end{bmatrix}$
- We partition the variables into P sets and split the constraints in the disjuction into P parts
- Next, we treat the splitted constraints as global constraints and take the convex hull of the disjunction, thus obtaining the P-split formulation

# Background – P-split formulation

$$\bigvee_{l \in \mathcal{D}} \left[ g_k(\boldsymbol{x}) \leq b_{l,k} \quad \forall k \in \mathcal{C}_l \right]$$
$$\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^n,$$

The general form of the disjunction

$$\alpha_s^l = \sum_{d \in \mathcal{D}} \nu_d^{\alpha_s^l} \qquad \forall\, s \in \{1, \dots, P\},\ \forall\, l \in \mathcal{D}$$

$$\sum_{s=1}^{P} \nu_l^{\alpha_s^l} \leq b_l \lambda_l \qquad \forall\, l \in \mathcal{D}$$

$$\underline{\alpha}_s^l \lambda_d \leq \nu_d^{\alpha_s^l} \leq \bar{\alpha}_s^l \lambda_d \qquad \forall\, s \in \{1, \dots, P\}, \forall\, l, d \in \mathcal{D}$$

$$\sum_{i \in \mathcal{I}_s} h_{i,l}(x_i) \leq \alpha_s^l \qquad \forall\, s \in \{1, \dots, P\},\ \forall\, l \in \mathcal{D}$$

$$\sum_{l \in \mathcal{D}} \lambda_l = 1, \quad \boldsymbol{\lambda} \in \{0, 1\}^{|\mathcal{D}|}$$

$$\boldsymbol{x} \in \mathcal{X}, \boldsymbol{\alpha}^l \in \mathbb{R}^P,\ \boldsymbol{\nu}^{\alpha_s^l} \in \mathbb{R}^{|\mathcal{D}|} \quad \forall\, s \in \{1, \dots, P\},\ \forall\, l \in \mathcal{D}\,,$$

P-split

Jan Kronqvist, Ruth Misener, Calvin Tsay, (2022). P-split formulations: A class of intermediate formulations between big-M and convex hull for disjunctive constraints

# Background – P-split formulation

- P-split is a formulation between convex hull and big-M: computationally light with a tight relaxation
- Assumptions:
  - 1. $g_k\colon \mathbb{R}^n \rightarrow \mathbb{R}$ are convex additively separable functions, i.e. $g_k(x) = \sum_{i=1}^{n} h_{ik}(x_i)$ where $h_{ik}\colon \mathbb{R} \rightarrow \mathbb{R}$ are convex
  - 2. All functions $g_k$ are bounded over $\chi$
  - 3. $|C_l| \ll n \ \forall l \in D$
- Properties:
  - The 1-split formulation is equal to the big-M formulation
  - A P+1-split formulation is always as tight or tighter than the corresponding P-split formulation.

# Objective

- Formulate a model which represents a trained ReLU neural network as a P-split formulation in Julia
- Computational experiments: comparing the ”classic” and P-split formulation
  - Should we use the P-split formulation?

# Tools

- Programming language: Julia
  - Neural networks: Flux.jl
  - 0-1 MILP formulation: Gogeta.jl
- Solver: Gurobi
- Data: Conrete Compressive Strength and possibly other datasets

# Schedule

- Introduction to the topic and Julia 01-02/2024
- Building the first NN and converting it to MIP using Gogeta.jl 02-03/2024
- Topic presentation 20.03.2024
- Formulating the code for P-split formulation 03-04/2024
- Writing the thesis 04-06/2024
- Thesis ready by 06/2024

# References and literature


- Matteo Fischetti, Jason Jo, (2018). Deep neural networks and mixed integer linear optimization
- Jan Kronqvist, Ruth Misener, Calvin Tsay, (2022). P-split formulations: A class of intermediate formulations between big-M and convex hull for disjunctive constraints
- Gogeta.jl: https://gamma-opt.github.io/Gogeta.jl/dev/