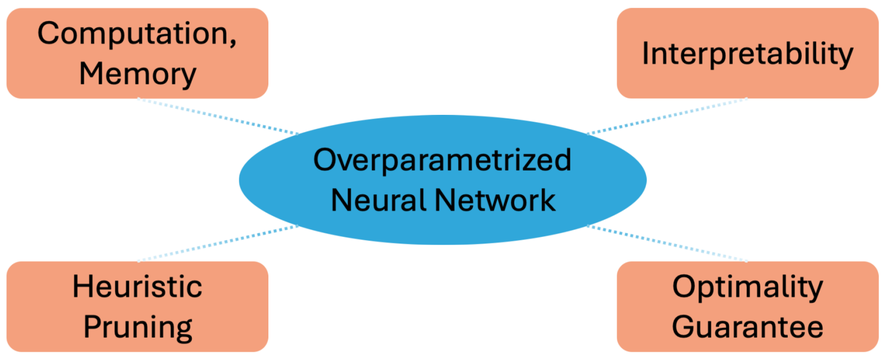
Sparsest Neural Network via MILP

Changwon Lee, Jeff Linderoth

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

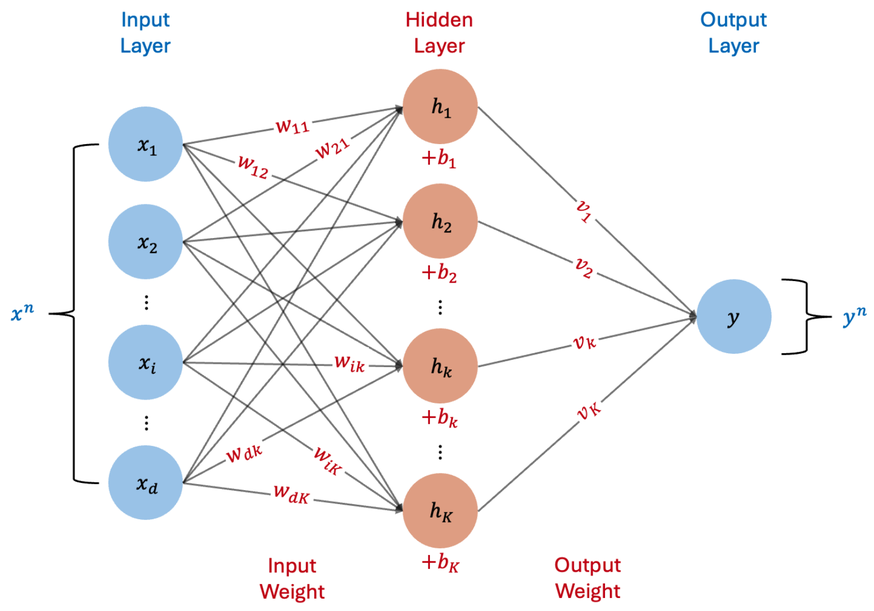
Motivation



- **Objective:** Find the *sparsest* single-hidden-layer ReLU network that fits all training data *exactly*
- **Sparsity:** Minimize the number of nonzero input–output weight products (ℓ_0 path norm)

Notation

Single-hidden-layer ReLU(σ) network:



$$f_{w,b,v}(\mathbf{x}) = \sum_{k=1}^{K} v_k\, \sigma\left(\sum_{i=1}^{d} w_{ik}\, x_i + b_k\right)$$

Input Data

- N data points $\{(x^n \in \mathbb{R}^d, y^n \in \mathbb{R})\}_{n=1}^{N}$

Optimization Problem (Nakhleh & Nowak)

$$\min_{w,b,v} \quad \sum_{k=1}^{K} \|v_k w_k\|_0 + \left(\sum_{k=1}^{K} \|v_k b_k\|_0\right)$$
$$\text{s.t.} \quad f_{w,b,v}(x^n) = y^n, \quad n = 1, \ldots, N$$
$$\|v_k w_k\|_\infty, \|v_k b_k\|_\infty \le R, \qquad \forall k$$

Decision Variables

- $w_{ik} \in [-\bar{w}, \bar{w}]$, $b_k \in [-\bar{b}, \bar{b}]$, $v_k \in [-\bar{v}, \bar{v}]$
- $p_k^n(+), q_k^n(-) \ge 0$: ReLU Output
- $z_k^n \in \{0,1\}$: ReLU Activation Indicator
- $u_{ik} = w_{ik} v_k$: ℓ_0 *path norm* (Input–Output Weight Product)
- $t_{ik}, s_k \in \{0,1\}$: Sparsity Indicators for w, b

Future Work

- **General Network Architecture:** Multivariate output y, multi-layer networks
- **Alternative Sparsity Objective:** Minimize number of active neurons, etc.
- **Optimal Branching Rule (L_k,t_{ik},s_k):** Exploit symmetry-breaking constraints in b&b
- **Parallel Computing (HPC):** Solve $K-1$ subproblems concurrently

Formulation

Baseline (MINLP-1) Fischetti & Jo (2018) | **Our Approach (MILP-3)** Symmetry Breaking

$$\min \quad \sum_{i=1}^{d}\sum_{k=1}^{K} t_{ik} + \sum_{k=1}^{K} s_k \quad \text{s.t.}$$

ReLU Output (Linear Combination)

$$\sum_{i=1}^{d} w_{ik}\, x_i^n + b_k = p_k^n - q_k^n \qquad \forall n, k$$

ReLU Activation (Big-M)

$$p_k^n, q_k^n \ge 0,\ z_k^n \in \{0,1\} \qquad \forall n, k$$
$$p_k^n \le M_n z_k^n,\ q_k^n \le M_n(1 - z_k^n)$$

Exact Data Interpolation (Bilinear)

$$\sum_{k=1}^{K} p_k^n\, v_k = y^n \qquad \forall n$$

Nonzero Weight (Path) Indicator

$$u_{ik} = w_{ik} v_k,\ o_k = b_k v_k \quad \textbf{(Bilinear)}$$
$$-\bar{u}\, t_{ik} \le u_{ik} \le \bar{u}\, t_{ik}, \quad t_{ik} \in \{0,1\}$$
$$-\bar{o}\, s_k \le o_k \le \bar{o}\, s_k, \quad s_k \in \{0,1\}$$

Weight Bound (from Benchmark)

$$-\bar{w} \le w_{ik} \le \bar{w}, -\bar{b} \le b_k \le \bar{b}, -\bar{v} \le v_k \le \bar{v}$$

Big-M Selection

$$M_n = \bar{w}\|x^n\|_1 + \bar{b} + \epsilon \qquad \forall n$$

ReLU Output (Linear Combination)

ReLU Activation (Big-M)

Binarize/Enumerate Output Weight

$$v_k \in \{-1, +1\} \qquad \forall k$$
$$v^{k'} = (\underbrace{-1, \ldots, -1}_{k'}, \underbrace{+1, \ldots, +1}_{K-k'})$$

Exact Data Interpolation (Linear)

$$\sum_{k=1}^{K} p_k^n\, v_k^{k'} = y^n \qquad \forall n$$

Nonzero Weight (Path) Indicator

$$-\bar{w}\, t_{ik} \le w_{ik} \le \bar{w}\, t_{ik}, \quad t_{ik} \in \{0,1\}$$
$$-\bar{b}\, s_k \le b_k \le \bar{b}\, s_k, \quad s_k \in \{0,1\}$$

Lexicographic Order of Neurons

$$L_k = s_k + \sum_{i=1}^{d} 2^i t_{ik} \qquad \forall k$$
$$L_1 \le \cdots \le L_{k'},\ L_{k'+1} \le \cdots \le L_K$$

Weight Bound

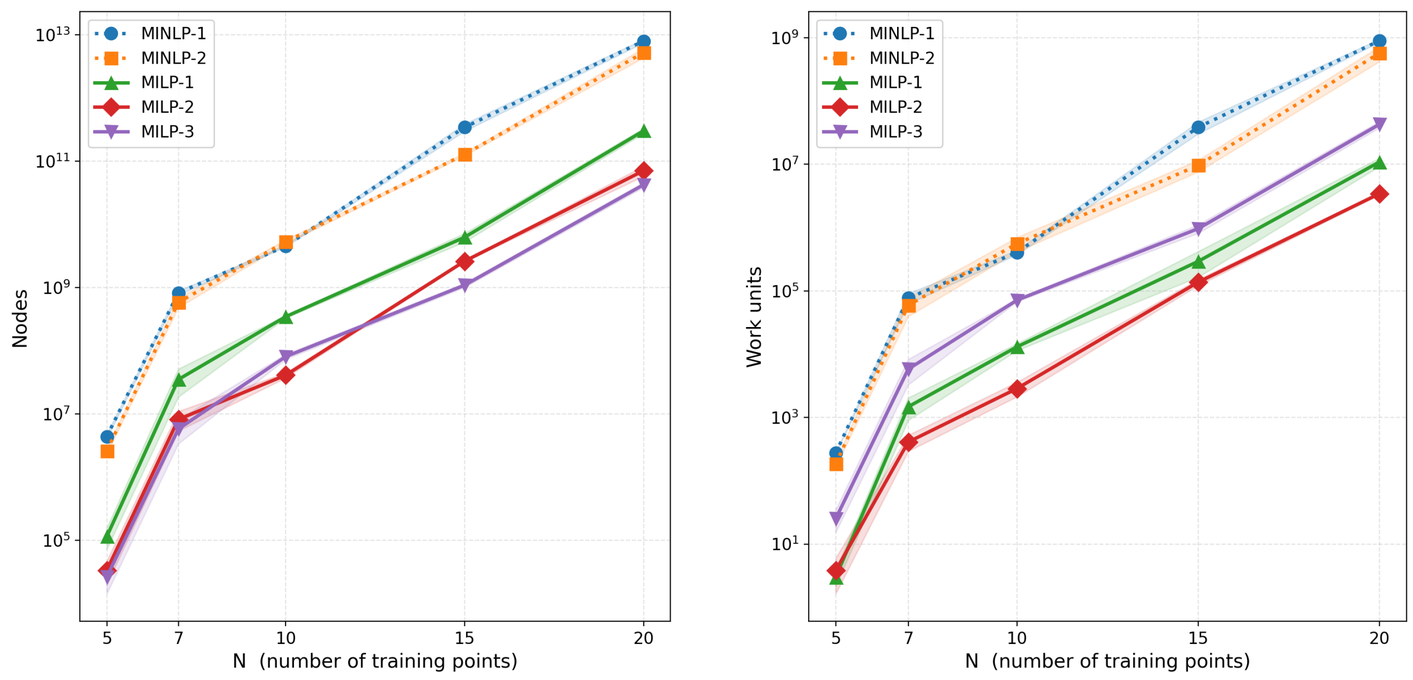
$$-\bar{w} \le w_{ik} \le \bar{w}, \quad -\bar{b} \le b_k \le \bar{b}$$

- **Reformulation (Linearization):** MINLP-1 → MINLP-2 → MILP-1
- **Neuron Symmetry Breaking:** MILP-1 → MILP-2 → MILP-3
- **MILP-2, 3:** Solve $K-1$ subproblems (one per sign pattern $v^{k'}$) → select best

Numerical Results

- **Dataset:** random, teacher network ($K = N/2$ neurons), low-rank subspace
- **Assumption:** $N \gg d$, $N \ge K$; bounds from benchmark
- **Solver:** Gurobi 13.0.2, 3-hour limit, 10 instances per (N, data-gen method)

5 MIP Formulations Comparison ($d = 3$, dgm=teacher)



B&B Nodes Explored | Gurobi Work Units

- **Sparsity:** **MINLP** models fail to find optimal solutions within the 3-hour limit even for small N, leaving large optimality gaps. **MILP** models solve to optimality on moderate-N instances, recovering networks at least as sparse as the **ℓ_p-regularized benchmark**.
- **MILP-1 vs. MILP-2:** Breaking output-neuron symmetry (1 large problem → $K-1$ small subproblems) yields an order-of-magnitude speedup.
- **MILP-2 vs. MILP-3:** Marginal improvement for small-to-moderate d; gains grow with d, possibly due to ties in the lexicographic score L_k.