A Library of Mirrors: Deep Neural Nets in Low Dimensions are Convex Lasso Models with Reflection Features

Emi Zeger EMIZEGER@STANFORD.EDU
Department of Electrical Engineering

Yifei Wang WANGYF18@STANFORD.EDU
Department of Electrical Engineering

Aaron Mishkin AMISHKIN@CS.STANFORD.EDU
Department of Computer Science

Tolga Ergen TERGEN@LGRESEARCH.AI
LG AI Research

Emmanuel Candès CANDES@STANFORD.EDU
Department of Statistics and Department of Mathematics

Mert Pilanci PILANCI@STANFORD.EDU
Department of Electrical Engineering

Stanford University, Stanford, CA 94305–2004, USA

Abstract

We prove that training neural networks on 1-D data is equivalent to solving a convex Lasso problem with a fixed, explicitly defined dictionary matrix of features. The specific dictionary depends on the activation and depth. We consider 2-layer networks with piecewise linear activations, deep narrow ReLU networks with up to 4 layers, and rectangular and tree networks with sign activation and arbitrary depth. Interestingly in ReLU networks, a fourth layer creates features that represent reflections of training data about themselves. The Lasso representation sheds insight to globally optimal networks and the solution landscape.

1 Introduction

Training deep neural networks is an important optimization problem. However, the non-convexity of neural nets makes their training challenging. We show that for low-dimensional data, e.g., 1-D or 2-D, training can be simplified to solving a convex Lasso problem with an easily constructable dictionary matrix.

Neural networks are used as predictive models for low-dimensional data in acoustic signal processing (Bianco et al., 2019; Freitag et al., 2017; Hsu and Jang, 2009; Mavaddati, 2020; Purwins et al., 2019; Serrà et al., 2019), physics-informed machine learning problems, uncertainty quantification (Chen and Ghattas, 2020; Chen et al., 2019; Stuart, 2014; Wang et al., 2022a,b; Zahm et al., 2022), and predicting financial data (Section 8).

In (Ergen and Pilanci, 2021a,b; Savarese et al., 2019), the problem of learning 1-D data is studied for two-layer ReLU networks, and it is proved that the optimal two-layer ReLU neural network precisely interpolates the training data as a piecewise linear function for which the breakpoints are at the data points. Recent work in (Joshi et al., 2023; Karhadkar

et al., 2023; Kornowski et al., 2023) also studied 2-layer ReLU neural networks and examined their behavior on 1-D data.

On the other hand, the current literature still lacks analysis on the expressive power and learning capabilities of deeper neural networks with generic activations. This motivates us to study the optimization of two-layer networks with piecewise linear activations and deeper neural networks with sign and ReLU activations. For 1-D data, we simplify the training problem by recasting it as a convex Lasso problem, which is well studied (Efron et al., 2004; Tibshirani, 1996, 2013).

Convex analysis of neural networks was studied in several prior works. As an example, infinite-width neural networks enable the convexification of the overall model (Bach, 2017; Bengio et al., 2005; Fang et al., 2019) . However, due to the infinite-width assumption, these results do not reflect finite-width neural networks in practice. Recently, a series of papers (Ergen and Pilanci, 2020, 2021a; Pilanci and Ergen, 2020) developed a convex analytic framework for the training problem of two-layer neural networks with ReLU activation. As a follow-up work, a similar approach is used to formulate the training problem for threshold activations in general $d \geq 1$ dimensions as a Lasso problem Ergen et al. (2023). However, the dictionary matrix is described implicitly and requires high computational complexity to create Ergen et al. (2023). By focusing on 1-D data, we provide simple, explicit Lasso dictionaries and consider additional activations, including sign activation, which are useful in many contexts such as saving memory to meet hardware constraints Bulat and Tzimiropoulos (2019); Kim and Smaragdis (2018).

Throughout this paper, all scalar functions extend to vector and matrix inputs componentwise. We denote vectors as $\mathbf{v} = (v_1, \cdots, v_n)$ and denote the set of column and row vectors by $\mathbb{R}^n$ and $\mathbb{R}^{1\times n}$, respectively. For $L \geq 2$, an L-layer *neural network* for d-dimensional data is denoted by $f_L(\mathbf{x}; \theta) : \mathbb{R}^{1\times d} \to \mathbb{R}$, where $\mathbf{x} \in \mathbb{R}^{1\times d}$ is an input row vector and $\theta \in \Theta$ is a *parameter set*. The set θ may contain matrices, vectors, and scalars representing weights and biases, and Θ is the *parameter space*. We let $\mathbf{X} \in \mathbb{R}^{N\times d}$ be a *data matrix* consisting of N *training samples* $\mathbf{x}^{(1)}, \dots \mathbf{x}^{(N)} \in \mathbb{R}^{1\times d}$, and let $\mathbf{y} \in \mathbb{R}^N$ be a *target vector*. We consider regression tasks, not classification. The (non-convex) neural net *training problem* is

$$\min_{\theta\in\Theta} \frac{1}{2}\|f_L(\mathbf{X}; \theta) - \mathbf{y}\|_2^2 + \frac{\beta}{\tilde{L}}\|\theta_w\|_{\tilde{L}}^{\tilde{L}} \tag{1}$$

where $\beta > 0$ is a regularization coefficient for a subset of parameters $\theta_w \subset \theta$ that incur a weight penalty when training. We denote $\|\theta_w\|_{\tilde{L}}^{\tilde{L}} = \sum_{q\in S_w} |q|^{\tilde{L}}$, where S_w is the set of elements of all matrices/vectors/scalars in θ_w. $\tilde{L}$ is the *effective regularized depth*, defined to be L for ReLU, leaky ReLU, and absolute value activations, and 2 for threshold and sign activations. The effective regularized depth captures the idea that unlike for other activations, for sign or threshold activation, only the weights of the final layer should be regularized, since all other weights are passed through an activation that is invariant to their magnitude (Remark 6, Appendix A.2).

In this paper we consider the *Lasso problem*

$$\min_{\mathbf{z},\xi} \frac{1}{2}\|\mathbf{A}\mathbf{z} + \xi\mathbf{1} - \mathbf{y}\|_2^2 + \beta\|\mathbf{z}\|_1 \tag{2}$$

where $\mathbf{z}$ is a vector, $\xi \in \mathbb{R}$, $\mathbf{1}$ is a vector of ones, and $\beta > 0$. $\mathbf{A}$ is called the *dictionary matrix* and is determined by the depth L and the activation of the neural net. The columns of $\mathbf{A}$ are *features* $\mathbf{A}_i \in \mathbb{R}^N$. The set of features is the *dictionary*. We call a collection of dictionaries for the same activation and different depths a *library*.

A neural net is *trained* by choosing θ that solves (1), and the resulting neural net is *optimal*. Unfortunately, this is complicated by the non-convexity of the neural net $f_L(\mathbf{x}, \theta)$ Pilanci and Ergen (2020). However, for data of dimension $d = 1$ we reformulate the training problem (1) into the equivalent but simpler Lasso problem (2), where $\mathbf{A}$ is a fixed matrix that is constructed based on the training data $\mathbf{X}$. We explicitly provide the elements of $\mathbf{A}$, making it straightforward to build and solve the convex Lasso problem instead of solving the non-convex problem (1). This reformulation allows for exploiting fast Lasso solvers such as Least Angle Regression (LARS) Efron et al. (2004).

Whereas in the training problem (1), the quality of the neural net fit to the data is measured by the l_2 loss as $\frac{1}{2}||f_L(\mathbf{X}; \theta) - \mathbf{y}||_2^2$, our results generalize to a wide class of convex loss functions $\mathcal{L}_{\mathbf{y}} : \mathbb{R}^N \to \mathbb{R}$. With a general loss function, (1) becomes $\min_{\theta \in \Theta} \mathcal{L}_{\mathbf{y}}(f_L(\mathbf{X}; \theta)) + (\beta/2)\|\theta_w\|_2^2$. This is shown to be equivalent to the generalization of (2), namely $\min_{\mathbf{z}, \xi} \mathcal{L}_{\mathbf{y}}(\mathbf{A}\mathbf{z} + \xi\mathbf{1} - \mathbf{y}) + \beta\|\mathbf{z}\|_1$.

The Lasso problem selects solutions $\mathbf{z}$ that generalize well by penalizing their total weight in l_1 norm Tibshirani (1996). The l_1 norm typically selects a minimal number of elements in $\mathbf{z}$ to be nonzero. The Lasso equivalence demonstrates that neural networks can learn a sparse representation of the data by selecting dictionary features to fit $\mathbf{y}$.

The Lasso representation also elucidates the solution path of neural networks. The *solution path* for the Lasso or training problem is the map from $\beta \in (0, \infty)$ to the solution set. The Lasso solution path is well understood (Tibshirani, 1996, 2013; Efron et al., 2004), providing insight into the solution path of the ReLU training problem (Mishkin and Pilanci, 2023).

This paper is organized as follows. We define the neural networks under consideration in Section 2. Section 3 describes our main theoretical result: neural networks are solutions to Lasso problems. Section 4 then examines the relationship between the entire set of optimal neural nets given by the training problem versus the Lasso problem. Section 5 applies our theory to examine neural net behavior under minimum regularization by studying the Lasso problem as $\beta \to 0$. Section 6 applies our theory to examples to explicitly find optimal neural networks. Section 7 presents experiments that support our theory in Section 3 and Section 5, and shows examples where neural networks trained with ADAM naturally exhibit Lasso features. Finally, Section 8 applies our theory to real-world data by predicting Bitcoin prices with neural networks and demonstrating improved performance by using the Lasso problem.

1.1 Contributions

We show the following main results.

- Training various neural network architectures on 1-D data is equivalent to solving Lasso problems with explicit, fixed and discrete dictionaries of basis signals that grow richer with depth (Theorems 1 and 2). We identify these dictionaries in closed form for ReLU and sign activations.

- Features with reflections of training data appear in the ReLU library at depth 4. In contrast, no reflection features are generated for the sign activation for any depth.

- Experimentally, training deep ReLU networks using the Adam optimizer leads to the same reflection features and matches our theoretical results on the global optima (Section 7).

- For certain binary classification tasks, the Lasso problem yields closed-form, optimal neural networks with sign activation. In such tasks, we analytically observe that 3-layer networks generalize better than 2-layer networks in that their predictions are more uniform (Corollaries 3 and 4)

- After depth 3, the sign activation library freezes for parallel neural networks but grows for tree-structured neural networks (Theorem 2).

- A similar convexification extends to 2-D data on the upper half plane (Theorem 3).

1.2 Notation

When the data dimension is $d = 1$, we assume $x_1 > x_2 > \cdots > x_N$. For a logical statement z, denote $\mathbf{1}\{z\}$ as its indicator function. For $n \in \mathbb{N}$, let $[n] = \{1, 2, \ldots, n\}$. For a matrix $\mathbf{Z}$, let $\mathbf{Z}_S$ be the submatrix of $\mathbf{Z}$ corresponding to indices in S. For a set of vectors $\mathcal{H}$, let $[\mathcal{H}]$ be a matrix whose columns are the elements of $\mathcal{H}$. For a vector $\mathbf{z}$, $||\mathbf{z}||_0$ is the number of nonzero elements in $\mathbf{z}$. Let $\mathbf{e}^{(n)} \in \mathbb{R}^N$ be the n^{th} canonical basis vector, that is $\mathbf{e}_i^{(n)} = \mathbf{1}\{i = n\}$. Let $\mathbf{1}_n, \mathbf{0}_n \in \mathbb{R}^n$ be the all-ones and all-zeros vectors, respectively, and without subscripts they are in $\mathbb{R}^N$.

2 Neural net architectures

This section is devoted to defining neural net terminology and notation to be used throughout the rest of the paper. Let $L \geq 2$ be the depth of a neural network ($L - 1$ hidden layers). The neural net activation $\sigma : \mathbb{R} \to \mathbb{R}$ is either the ReLU $\sigma(x) = (x)_+ := \max\{x, 0\}$, absolute value $\sigma(x) = |x|$, leaky ReLU, threshold or sign function. For fixed slopes $a, b \in \mathbb{R}$, the leaky ReLu is $\sigma(x) = (a\mathbf{1}\{x > 0\} + b\mathbf{1}\{x < 0\})x$. The threshold activation is $\sigma(x) = \mathbf{1}(x)$, where $\mathbf{1}(x) = \mathbf{1}\{x \geq 0\}$, and the sign activation is $\sigma(x) = \text{sign}(x)$, where $\text{sign}(x)$ is -1 if $x < 0$, and 1 if $x \geq 0$. Note $\text{sign}(0) = 1$. For $\mathbf{Z} \in \mathbb{R}^{n \times m}, \mathbf{s} \in \mathbb{R}^m$, let $\sigma_{\mathbf{s}}(\mathbf{Z}) = \sigma(\mathbf{Z})\text{Diag}(\mathbf{s})$. When each column of $\sigma(\mathbf{Z})$ is a neuron output, each column of $\sigma_{\mathbf{s}}(\mathbf{Z}) \in \mathbb{R}^{N \times m}$ is a neuron scaled by an *amplitude parameter* $s_i \in \mathbb{R}$. Amplitude parameters are (trainable) parameters for sign and threshold activations, and ignored (even if written) for ReLU, leaky ReLU, and absolute value activations.

Next we define some neural net architectures. The parameter set is partitioned into $\theta = \theta_w \cup \theta_b \cup \{\xi\}$, where θ_b is a set of *internal bias* terms, and ξ is an *external* bias term. We will define the elements of each parameter set below. We will define neural nets by their output on row vectors $\mathbf{x} \in \mathbb{R}^{1 \times d}$. Their outputs then extend to matrix inputs $\mathbf{X} \in \mathbb{R}^{N \times d}$ row-wise.

2.1 Standard networks

The following is a commonly studied neural net architecture. Let $L \geq 2$, the number of layers. Let $m_0 = d, m_{L-1} = 1$ and $m_l \in \mathbb{N}$ for $l \in [L-2]$, which are the number of neurons in each layer. For $l \in [L-1]$, let $\mathbf{W}^{(l)} \in \mathbb{R}^{m_{l-1} \times m_l}, \mathbf{s}^{(l)} \in \mathbb{R}^{m_l}, \mathbf{b}^{(l)} \in \mathbb{R}^{1 \times m_l}, \xi \in \mathbb{R}$, which are the weights, amplitude parameters, internal biases, and external bias, respectively. Let $\mathbf{X}^{(1)} = \mathbf{x} \in \mathbb{R}^{1 \times d}$ be the input to the neural net and $\mathbf{X}^{(l+1)} \in \mathbb{R}^{1 \times m_l}$ be viewed as the inputs to layer $l+1$, defined by

$$\mathbf{X}^{(l+1)} = \sigma_{\mathbf{s}^{(l)}} \left(\mathbf{X}^{(l)} \mathbf{W}^{(l)} + \mathbf{b}^{(l)} \right). \tag{3}$$

Let $\boldsymbol{\alpha} \in \mathbb{R}^{m_L}$, which is the vector of final layer coefficients. A *standard neural network* is $f_L(\mathbf{x}; \theta) = \xi + \mathbf{X}^{(L)} \boldsymbol{\alpha}$. The regularized and bias parameter sets are $\theta_w = \left\{ \boldsymbol{\alpha}, \mathbf{W}^{(l)}, \mathbf{s}^{(l)} : l \in [L-1] \right\}$ and $\theta_b = \left\{ \mathbf{b}^{(l)} : l \in [L-1] \right\}$.

There is much interest in analyzing the training problem for standard networks, but this appears to be a challenging problem. However, by changing the architecture to a parallel or tree structure defined below, we show that the training problem simplifies to the Lasso problem. These alternative architectures allow neural nets to be reconstructed more tractably from a Lasso solution than with a standard network. In the parallel and tree architectures, m_L is the number of neurons in the final layer and for $i \in [m_L]$, we define the disjoint unions $\theta_w = \bigcup_{i \in [m_L]} \theta_w^{(i)}$ and $\theta_b = \bigcup_{i \in [m_L]} \theta_b^{(i)}$.

2.2 Parallel networks

A parallel network is a linear combination of standard networks in parallel, as we now define. Each standard network is called a *parallel unit*. Let $L \geq 2, m_0 = d, m_{L-1} = 1$ and $m_l \in \mathbb{N}$ for $l \in [L] - \{L-1\}$. For $i \in [m_L], l \in [L-1]$, let $\mathbf{W}^{(i,l)} \in \mathbb{R}^{m_{l-1} \times m_l}$, $\mathbf{s}^{(i,l)} \in \mathbb{R}^{m_l}$, $\mathbf{b}^{(i,l)} \in \mathbb{R}^{1 \times m_l}, \xi \in \mathbb{R}$, which are the weights, amplitude parameters, and biases of the i^{th} parallel unit. Let $\mathbf{X}^{(i,1)} = \mathbf{x} \in \mathbb{R}^{1 \times d}$ be the input to the neural net and $\mathbf{X}^{(i,l+1)} \in \mathbb{R}^{1 \times m_l}$ be viewed as the input to layer $l+1$ in unit i, defined by

$$\mathbf{X}^{(i,l+1)} = \sigma_{\mathbf{s}^{(i,l)}} \left(\mathbf{X}^{(i,l)} \mathbf{W}^{(i,l)} + \mathbf{b}^{(i,l)} \right). \tag{4}$$

Let $\boldsymbol{\alpha} \in \mathbb{R}^{m_L}$. A *parallel neural network* is $f_L(\mathbf{x}; \theta) = \xi + \sum_{i=1}^{m_L} \alpha_i \mathbf{X}^{(i,L)}$. The regularized and bias parameter sets are $\theta_w^{(i)} = \left\{ \alpha_i, \mathbf{s}^{(i,l)}, \mathbf{W}^{(i,l)} : l \in [L-1] \right\}, \theta_b^{(i)} = \left\{ \mathbf{b}^{(i,l)} : l \in [L-1] \right\}$, for $i \in [m_L]$. A *deep narrow network* is a parallel neural net with $m_1 = \cdots = m_{L-1} = 1$. For $L \geq 3$, a *rectangular network* is a parallel network with $m_1 = \cdots = m_{L-2}$.

2.3 Tree networks

Let $L \geq 3, m_2, \cdots, m_L \in \mathbb{N}$. Given $l \in \{0, \cdots, L-2\}$, let $\mathbf{u}$ be an l-tuple where if $l = 0$, we denote $\mathbf{u} = \emptyset$ and otherwise, $\mathbf{u} = (u_1, \cdots, u_l)$ such that $u_i \in [m_{L-i}]$ for $i \in [l]$. For an integer a, denote $\mathbf{u} \oplus a$ as the concatenation $(u_1, \cdots, u_l, a)$. For $l \in [L-1]$, and $\mathbf{u}$ of length l, let $\alpha^{(\mathbf{u})}, s^{(\mathbf{u})}, b^{(\mathbf{u})}, \mathbf{w}^{(\mathbf{u})} \in \mathbb{R}$, except let $\mathbf{w}^{(u_1, \cdots, u_{L-1})} \in \mathbb{R}^d$. For all $\mathbf{u}$ of length $L-1$, let

$\mathbf{X}^{(u_1,\cdots,u_{L-1})} = \mathbf{x} \in \mathbb{R}^{1\times d}$. For $\mathbf{u}$ of length $l \in \{0,\cdots,L-2\}$, let $\mathbf{X}^{(\mathbf{u})} \in \mathbb{R}$ be defined by

$$\mathbf{X}^{(\mathbf{u})} = \sum_{i=1}^{m_{L-l}} \alpha^{(\mathbf{u}\oplus i)} \sigma_{s^{(\mathbf{u}\oplus i)}} \left(\mathbf{X}^{(\mathbf{u}\oplus i)} \mathbf{w}^{(\mathbf{u}\oplus i)} + b^{(\mathbf{u}\oplus i)} \right). \tag{5}$$

A *tree neural network* is $f_L(\mathbf{x};\theta) = \xi + \mathbf{X}^{(\emptyset)}$. Visualizing the neural network as a tree, $\mathbf{X}^{(\emptyset)}$ is the "root," $\mathbf{u} = (u_1,\cdots u_l)$ specifies the path from the root at level 0 to the $u_l{}^{\text{th}}$ node (or neuron) at level l, $\mathbf{X}^{(\mathbf{u})}$ represents a subtree at this node, and (5) specifies how this subtree is built from its child nodes $\mathbf{X}^{(\mathbf{u}\oplus i)}$. The leaves of the tree are all copies of $\mathbf{X}^{(u_1,\cdots,u_{L-1})} = \mathbf{X}$. Let $\mathcal{U} = \prod_{l=0}^{L-2}[m_{L-l}]$. The regularized and bias parameter sets are $\theta_w^{(i)} = \left\{\alpha^{(\mathbf{u})}, s^{(\mathbf{u})}, \mathbf{w}^{(\mathbf{u})} : \mathbf{u}\in\mathcal{U}, u_1 = i\right\}, \theta_b^{(i)} = \left\{b^{(\mathbf{u})} : \mathbf{u}\in\mathcal{U}, u_1 = i\right\}$. For tree networks, let $\boldsymbol{\alpha} = \left(\alpha^{(1)},\cdots,\alpha^{(m_L)}\right) \in \mathbb{R}^{m_L}$.

This paper primarily focuses on the parallel architecture. A parallel network can be converted into a standard network as follows. Let $\mathbf{W}^{(1)} = \left[\mathbf{W}^{(1,1)}\cdots\mathbf{W}^{(m_1,1)}\right]$. For $l \geq 1$, let $\mathbf{b}^{(l)} = \left(\mathbf{b}^{(1,l)}\cdots\mathbf{b}^{(m_l,l)}\right)$. For $l > 1$, let $\mathbf{W}^{(l)} = \text{blockdiag}\left(\mathbf{W}^{(1,l)}\cdots\mathbf{W}^{(m_l,l)}\right)$. And let $\boldsymbol{\alpha}, \xi$ be the same in the standard network as the parallel one.

While each unit of the parallel neural network is a standard network, every branch of the tree network is a parallel network. Standard and parallel nets have the same architecture for $L = 2$ layers, and parallel and tree nets are the same for $L = 3$ layers. For $L \geq 4$ layers, the architectures all diverge.

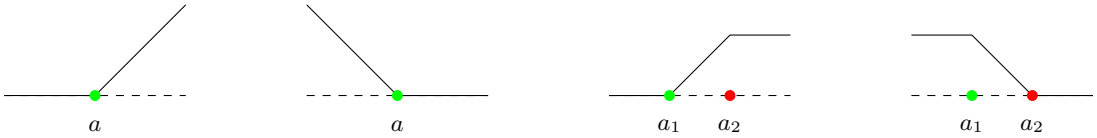
For all architectures, define *parameter unscaling* as follows. Parameter unscaling for ReLU, leaky ReLU, or absolute value activation is the transformation $q \to \text{sign}(q)\gamma_i$ for $q \in \theta_w^{(i)}$, and $q \to q\gamma_i$ for $q \in \theta_b^{(i)}$, where $\gamma_i = |\alpha_i|^{\frac{1}{L}}$. For sign and threshold activation, it is the transformation $\alpha_i \to \text{sign}(\alpha_i)\sqrt{|\alpha_i|}$ and $s^{(i,L-1)} \to \sqrt{|\alpha_i|}$ for parallel nets, and $s^{(i)} \to \sqrt{|\alpha_i|}$ for tree nets. This will be used in reconstructing neural nets from Lasso solutions.
Henceforth, except for Section 3.2.1 and the Appendix, assume the data is in 1-D.

3 Main results

In this section, we show that non-convex deep neural net training problems are *equivalent* to Lasso problems, that is, their optimal values are the same, and given a Lasso solution, we can reconstruct a neural net that is optimal in the training problem.

3.1 Deep narrow networks

We reformulate the training problem for 2-layer networks with piecewise linear activations and deeper networks with ReLU activation. Proofs are deferred to Appendix C. For a piecewise linear function $f : \mathbb{R} \to \mathbb{R}$, x is a *breakpoint* if f changes slope at x. We next define some parameterized families of functions from $\mathbb{R}$ to $\mathbb{R}$.



$\mathrm{ReLU}_a^+(x)$ $\mathrm{ReLU}_a^-(x)$ $\mathrm{Ramp}_{a_1,a_2}^+(x)$ $\mathrm{Ramp}_{a_1,a_2}^-(x)$

Figure 1: Examples of capped ramp functions in Definition 1.

Definition 1 Let $a_1 \in [-\infty, \infty), a_2 \in (-\infty, \infty]$. The *capped ramp* functions are

$$\mathrm{Ramp}_{a_1,a_2}^+(x) = \begin{cases} 0 & \text{if } x \le a_1 \\ x - a_1 & \text{if } a_1 \le x \le a_2 \\ a_2 - a_1 & \text{if } x \ge a_2 \end{cases}, \quad \mathrm{Ramp}_{a_1,a_2}^-(x) = \begin{cases} a_2 - a_1 & \text{if } x \le a_1 \\ a_2 - x & \text{if } a_1 \le x \le a_2 \\ 0 & \text{if } x \ge a_2, \end{cases}$$

provided that $a_1 \le a_2$, and otherwise $\mathrm{Ramp}_{a_1,a_2}^+ = \mathrm{Ramp}_{a_1,a_2}^- = 0$. In particular, the *ramp* functions are $\mathrm{ReLU}_a^+(x) = \mathrm{Ramp}_{a,\infty}^+ = (x-a)_+, \mathrm{ReLU}_a^-(x) = \mathrm{Ramp}_{-\infty,a}^- = (a-x)_+$.

In Definition 1, the parameters a, a_1, a_2 are the breakpoints of ramp and capped ramp functions. This is illustrated in Figure 1.

Definition 2 For $a, b \in \mathbb{R}$, the *reflection of a about b* is the point $R_{(a,b)} = 2b - a$.

For $a_1, a_2 \in \{x_1, \cdots, x_N\}$, the vectors $\mathrm{ReLU}_{a_1}^+(\mathbf{X})$ and $\mathrm{ReLU}_{a_1}^-(\mathbf{X})$ are called *ramp features* while $\mathrm{Ramp}_{a_1,a_2}^+(\mathbf{X})$ and $\mathrm{Ramp}_{a_1,a_2}^-(\mathbf{X})$ are *capped ramp features*. The vectors $\mathrm{ReLU}_{a_1}^+(\mathbf{X})$, $\mathrm{ReLU}_{a_1}^+(\mathbf{X}), \mathrm{Ramp}_{a_1,a_2}^+(\mathbf{X})$ and $\mathrm{Ramp}_{a_1,a_2}^-(\mathbf{X})$ are *reflection features* if $a_1, a_2 \in \{x_1, \cdots, x_N\} \cup \left\{R_{(x_{j_1}, x_{j_2})} : j_1, j_2 \in [N]\right\}$ and a_1 or a_2 is in $\left\{R_{(x_{j_1}, x_{j_2})} : j_1, j_2 \in [N]\right\}$. Using these features, we informally state our main result on Lasso equivalence for ReLU networks.

Theorem 1 (Informal) *A deep narrow network with ReLU activation of depth $2, 3, 4$ is equivalent to a Lasso model with ramp, capped ramp, and reflection features, respectively.*

We state Theorem 1 formally later in this section. The theorem suggests that depending on the depth, a ReLU network learns to model data with a discrete and fixed dictionary of features. Moreover, it suggests that as the depth increases, this dictionary expands, which deepens its representation power.

Remark 1 *Note that when the network depth is 2 or 3, the equivalent Lasso dictionary only contains capped ramp features with breakpoints at training data, leading to a prediction with kinks only at data locations. In contrast, when the network depth is 4, there can be breakpoints at **reflections** of data points with respect to other data points due to the reflection features. As a result, the sequence of dictionaries as the network gets deeper converges to a richer library that includes reflections.*

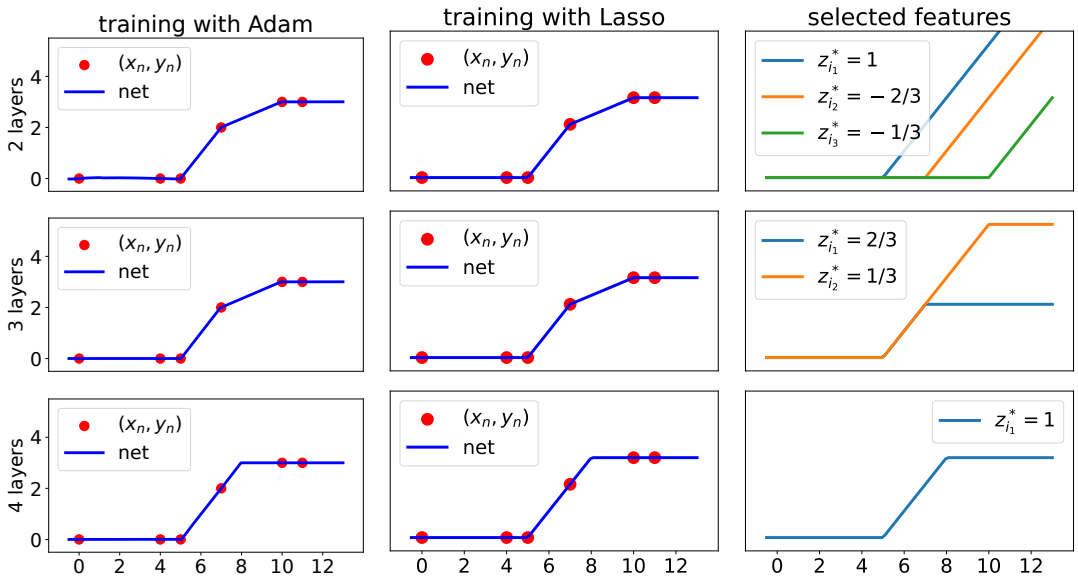


Figure 2: Plots of deep narrow ReLU network predictions (blue) for the same 1-D dataset (red dots), found by (left): training with Adam on the non-convex training problem and $\beta \approx 0$, (middle): analytically solving the minimum norm convex Lasso problem. Features corresponding to nonzero Lasso solution components z_i^* are plotted in the right column.

Definition 3 For an activation σ and $L \in \mathbb{N}$, define the following *dictionary index sets*:

$$\mathcal{M}^{(1)} = \begin{cases} \{1\} & \text{if } L = 2 \text{ and } \sigma(x) = \text{sign}(x) \text{ or } \sigma(x) = |x| \\ \{-1, 1\}^{L-1} & \text{else} \end{cases}, \qquad \mathcal{M}^{(2)} = [N]^{L-1},$$

$$\mathcal{M}^{(3)} = \begin{cases} \{0\} & \text{if } L < 4 \\ \{0, 1\} & \text{else} \end{cases}, \qquad \mathcal{M} = \mathcal{M}^{(1)} \times \mathcal{M}^{(2)} \times \mathcal{M}^{(3)}.$$

Recall that $\mathbf{s}^{(l)} \in \mathbb{R}^{m_l}, \mathbf{s}^{(i,l)} \in \mathbb{R}^{m_l}$ and $s^{(\mathbf{u})} \in \mathbb{R}$ denote amplitude parameters for standard, parallel and tree architectures, respectively. In this subsection (3.1), with slight abuse of notation, we denote elements of $\mathcal{M}^{(1)}$ by $\mathbf{s} = (s_1, \cdots, s_{L-1}) \in \{-1, 1\}^{L-1}$.

Definition 4 Let $L \in \{2, 3, 4\}, (\mathbf{s}, \mathbf{j}, k) \in \mathcal{M}$. The *deep narrow function* $f^{(L,\sigma)}_{\mathbf{s},\mathbf{j},k} : \mathbb{R} \to \mathbb{R}$ is defined as follows. Let $a_1 = x_{j_1}$ if $k = 0$ and otherwise let $a_1 = R_{(x_{j_1}, x_{j_2})}$.

If L = 2:

$$f^{(L,\sigma)}_{\mathbf{s},\mathbf{j},k}(x) = \begin{cases} \sigma(x_{j_1} - x) & \text{if } s_1 = -1 \\ \sigma(x - x_{j_1}) & \text{if } s_1 = 1 \end{cases}$$

If L = 3:

if $s_2 = 1$:

$$f^{(L,\sigma)}_{\mathbf{s},\mathbf{j},k}(x) = \begin{cases} \text{ReLU}^-_{\min\{x_{j_1}, x_{j_2}\}}(x) & \text{if } s_1 = -1 \\ \text{ReLU}^+_{\max\{x_{j_1}, x_{j_2}\}}(x) & \text{if } s_1 = 1 \end{cases}$$

if $s_2 = -1$:

$$f^{(L,\sigma)}_{\mathbf{s},\mathbf{j},k}(x) = \begin{cases} \text{Ramp}^{+}_{x_{j_2},x_{j_1}}(x) & \text{if } s_1 = -1 \\ \text{Ramp}^{-}_{x_{j_1},x_{j_2}}(x) & \text{if } s_1 = 1. \end{cases}$$

If L = 4:

if $s_2 = s_3 = 1$:

$$f^{(L,\sigma)}_{\mathbf{s},\mathbf{j},k}(x) = \begin{cases} \text{ReLU}^{-}_{\min\{a_1,x_{j_2},x_{j_3}\}}(x) & \text{if } s_1 = -1 \\ \text{ReLU}^{+}_{\max\{a_1,x_{j_2},x_{j_3}\}}(x) & \text{if } s_1 = 1 \end{cases}$$

if $s_2 = 1, s_3 = -1$:

$$f^{(L,\sigma)}_{\mathbf{s},\mathbf{j},k}(x) = \begin{cases} \text{Ramp}^{+}_{x_{j_3},\min\{a_1,x_{j_2}\}}(x) & \text{if } s_1 = -1 \\ \text{Ramp}^{-}_{\max\{a_1,x_{j_2}\},x_{j_3}}(x) & \text{if } s_1 = 1 \end{cases}$$

if $s_2 = -1, s_3 = 1$:

$$f^{(L,\sigma)}_{\mathbf{s},\mathbf{j},k}(x) = \begin{cases} \text{Ramp}^{+}_{\max\{x_{j_2},x_{j_3}\},a_1}(x) & \text{if } s_1 = -1 \\ \text{Ramp}^{-}_{a_1,\min\{x_{j_2},x_{j_3}\}}(x) & \text{if } s_1 = 1 \end{cases}$$

if $s_2 = -1, s_3 = -1$:

$$f^{(L,\sigma)}_{\mathbf{s},\mathbf{j},k}(x) = \begin{cases} \text{Ramp}^{-}_{x_{j_2},\min\{a_1,x_{j_3}\}}(x) & \text{if } s_1 = -1 \\ \text{Ramp}^{+}_{\max\{a_1,x_{j_3}\},x_{j_2}}(x) & \text{if } s_1 = 1. \end{cases}$$

We call $f^{(L,\sigma)}_{\mathbf{s},\mathbf{j},k}(\mathbf{X})$ a *deep narrow feature.*

Definition 5 Let Ramp_{a_1,a_2} be $\text{Ramp}^{+}_{a_1,a_2}$ if $a_1 \leq a_2$ and $\text{Ramp}^{-}_{a_2,a_1}$ otherwise.

Definition 5 states that $\text{Ramp}_{a_1,a_2}(x)$ is the capped ramp function with breakpoints at a_1, a_2 such that $\text{Ramp}_{a_1,a_2}(a_1) = 0$. When $L = 4$, deep narrow features can have breakpoints at reflections. However, not all reflections of data points are breakpoints, as described in the next result. For a ramp, its slope must be in the direction of the reflection: if $x_{j_2} > x_{j_1}$ then $R_{(x_{j_1},x_{j_2})} > x_{j_2}$ and a ramp must increase for $x > R_{(x_{j_1},x_{j_2})}$. If $x_{j_2} < x_{j_1}$ then $R_{(x_{j_1},x_{j_2})} < x_{j_2}$ and a ramp must decreases for $x < R_{(x_{j_1},x_{j_2})}$. For a capped ramp, at most one breakpoint can be a reflection. If a capped ramp evaluates to 0 at a data point breakpoint and the other breakpoint is a reflection $R_{(x_{j_1},x_{j_2})}$, then x_{j_1} has to reflect across x_{j_2} towards the breakpoint that is a data point. For example, an increasing capped ramp has a reflection breakpoint satisfying $R_{(x_{j_1},x_{j_2})} < x_{j_2} < x_{j_1}$. If the capped ramp has a value of 0 at a reflection breakpoint $R_{(x_{j_1},x_{j_2})}$, then the other breakpoint must be x_{j_2}. See Figure 3.

Lemma 1 *All $L = 4$ deep narrow functions with breakpoints at a reflection are of the form $ReLU^{+}_{R_{(x_{j_1},x_{j_2})}}$ if $j_1 \geq j_2$, $ReLU^{-}_{R_{(x_{j_1},x_{j_2})}}$ if $j_1 \leq j_2$, $Ramp_{x_{j_3},R_{(x_{j_1},x_{j_2})}}$ where $j_1 < j_2 \leq j_3$ or $j_1 > j_2 \geq j_3$, and $Ramp_{R_{(x_{j_1},x_{j_2})},x_{j_2}}$ where $j_1 \neq j_2$.*

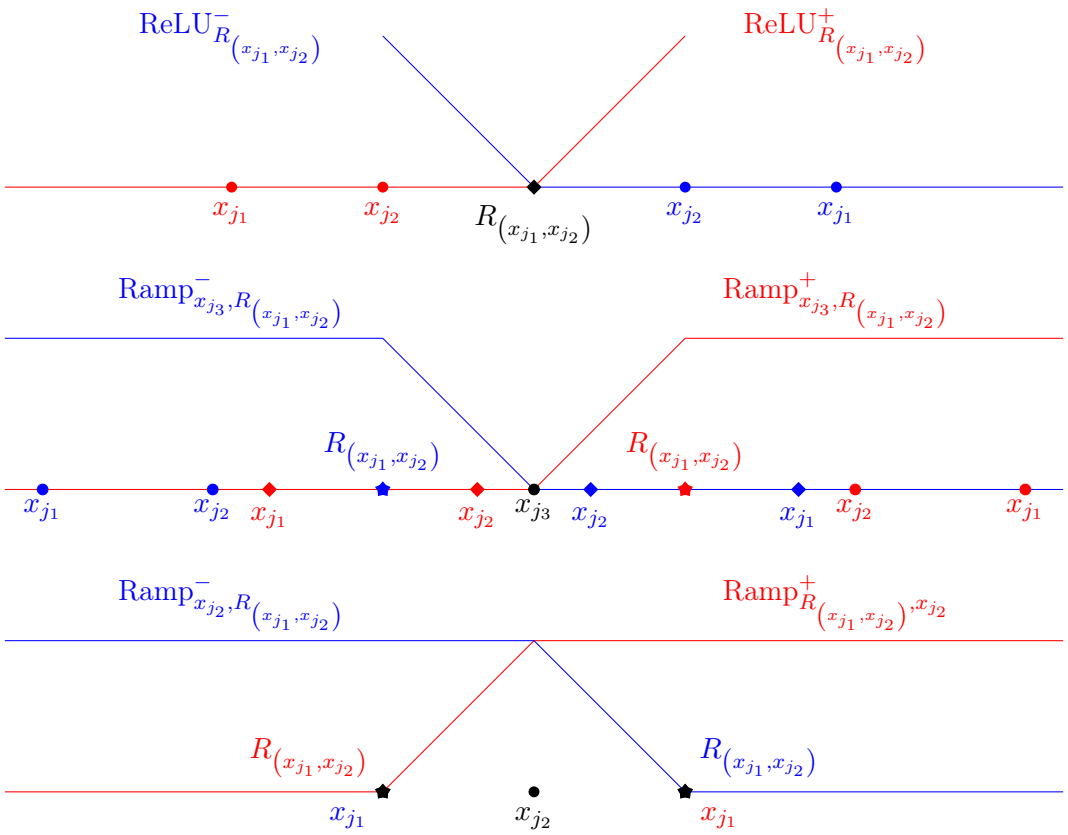


Figure 3: $L = 4$ deep narrow features that have a breakpoint at a reflection point. Top: ramp features. Middle: $\text{Ramp}_{x_{j_3},R_{(x_{j_1},x_{j_2})}}$. Bottom: $\text{Ramp}_{R_{(x_{j_1},x_{j_2})},x_{j_2}}$.

The next result shows the training problem (1) for a deep narrow network (defined in Section 2.2) can be optimized via a Lasso problem (2) that learns deep narrow features $f^{(L,\sigma)}_{\mathbf{s},\mathbf{j},k}(\mathbf{X})$ for different tuples of parameters $(\mathbf{s},\mathbf{j},k) \in \mathcal{M}$.

Theorem 1 *Let $L \in \{2,3,4\}$. Let σ be the ReLU, leaky ReLU, absolute value, sign, or threshold function if $L = 2$, and ReLU otherwise. Consider a Lasso problem whose dictionary consists of all possible deep narrow features and where $\xi = 0$ if σ is the sign or threshold function. Suppose $(\mathbf{z}^*, \xi^*)$ is a solution, and let $m^* = \|\mathbf{z}^*\|_0$. This Lasso problem is equivalent to a training problem for a L-layer deep narrow network with activation σ and $m_L \geq m^*$.*

The notion of equivalence between optimization problems is defined in the beginning of Section 3. Theorem 1 shows that instead of training a neural network with a non-convex problem and reaching a possibly local optimum, we can simply solve a straightforward Lasso problem whose convexity guarantees that gradient descent approaches global optimality. For ReLU activation, the dictionary matrix has up to $|\mathcal{M}^{(1)}| \cdot |\mathcal{M}^{(2)}| \cdot |\mathcal{M}^{(3)}|$ features, which is $2^{L-1}N^{L-1}$ for depth $L = 2, 3$ and $2^L N^{L-1}$ for $L = 4$. In previous work (Ergen et al., 2023), a similar Lasso formulation is developed for networks with threshold activation but requires

up to 2^N features of length N in the dictionary for a 2-layer network. In contrast, Theorem 1 shows that at most $2N^2$ features are needed for a 2-layer network.

Theorem 1 states how the Lasso dictionary evolves with depth, adding more features with each layer. A neural net trained with the Lasso problem in Theorem 1 learns dictionary functions $f^{(L,\sigma)}_{\mathbf{s},\mathbf{j},k}(\mathbf{X})$ as features, which are ramps for $L = 2$ and capped ramps for $L > 2$, sampled at the training data. For $L = 2, 3$, the features have breakpoints at data points, and for $L = 4$, also at their reflections. The data and reflection breakpoints correspond to the cases $a_1 = x_{j_1}$ and $a_1 = R_{(x_{j_1}, x_{j_2})}$, respectively, in Definition 4. Figure 1 illustrates the basic types of features. Figure 5 enumerates Lasso features for each depth.

Remark 2 *Let $L = 2$ and $\mathbf{A}_+, \mathbf{A}_- \in \mathbb{R}^{N\times N}$ with elements $(\mathbf{A}_+)_{i,n} = \sigma(x_i - x_n), (\mathbf{A}_-)_{i,n} = \sigma(x_i - x_n)$. We can write the dictionary matrix in Theorem 1 as $\mathbf{A} = \mathbf{A}_+$ for absolute value and sign activations, and $\mathbf{A} = [\mathbf{A}_+, \mathbf{A}_-] \in \mathbb{R}^{N\times 2N}$ for ReLU, leaky ReLU, and threshold activations. $\mathbf{A}_+$ and $\mathbf{A}_-$ contain features $f^{(L,\sigma)}_{\mathbf{s},\mathbf{j},k}(\mathbf{X})$ where $\mathbf{s} = 1$ and $\mathbf{s} = -1$, respectively.*

Figure 4 illustrates $\mathbf{A}_+$ for the ReLU activation. Next, we discuss a map to reconstruct an optimal neural net from a Lasso solution. As defined in Definition 4, the deep narrow features are specified by the tuples $i = (\mathbf{s}, \mathbf{j}, k) \in \mathcal{M}$. Hence for deep narrow networks we also index the columns of $\mathbf{A}$, the elements of vector $\mathbf{z}$ in the Lasso problem, and the reconstructed parallel units by the tuples $i \in \mathcal{M}$. Note for a deep narrow network, $\mathbf{W}^{(i,l)}, \mathbf{b}^{(i,l)} \in \mathbb{R}$.

Definition 6 Let $(\mathbf{z}^*, \xi^*)$ be a solution to the Lasso problem. The *reconstructed parameters* for a deep narrow network are defined as follows. For $i = (\mathbf{s}, \mathbf{j}, k) \in \mathcal{M}$, let $a_1^{(i)}$ be a_1 as defined in Definition 4, and let $a_2^{(i)} = \left(s_1\left(x_{j_2} - a_1^{(i)}\right)\right)_+$ if $L > 2$, $a_3^{(i)} = \left(s_2\left(\left(s_1\left(x_{j_3} - a_1^{(i)}\right)\right)_+ - a_2^{(i)}\right)\right)_+$ if $L > 3$. Let $\boldsymbol{\alpha} = \mathbf{z}^*$ and $\xi = \xi^*$. For sign and threshold activation, let all amplitude parameters be 1. For $i = (\mathbf{s}, \mathbf{j}, k) \in \mathcal{M}, l \in [L-1]$, let $\mathbf{W}^{(i,l)} = s_l$, $\mathbf{b}^{(i,l)} = -s_l a_l^{(i)}$. Finally, unscale parameters (as defined in Section 2).

A reconstructed deep narrow network is optimal in the training problem, as shown in the proof of Theorem 1. The reconstruction is efficient and explicit. Next, we simplify Definition 6 for shallow networks. For $L = 2$, $\mathcal{M} = \{-1, 1\} \times [N] \times \{0\}$ for ReLU, leaky ReLU, and threshold activations and $\mathcal{M} = \{1\} \times [N] \times \{0\}$ for sign and absolute value activations, and the layer index is $l \in [L-1] = \{1\}$.

Definition 7 For $L = 2$ and $i = (\mathbf{s}, \mathbf{j}, k) \in \mathcal{M}$, define the scalars $w_i = \mathbf{W}^{(i,1)}, b_i = \mathbf{b}^{(i,1)}, \tilde{\alpha}_i = \mathbf{s}|\alpha_i|$. Let $\mathbf{w}, \mathbf{b}, \tilde{\boldsymbol{\alpha}}$ be vectors stacking together all $w_i, b_i, \tilde{\alpha}_i$, respectively.

In the following, take vector-vector operations elementwise.

Remark 3 *Let $R^{z\to\alpha}(\mathbf{z}) = sign(|\mathbf{z}|)\sqrt{|\mathbf{z}|}$. Let $R^{\alpha,\xi\to\theta}(\boldsymbol{\alpha}, \xi) = (\boldsymbol{\alpha}, \xi, \tilde{\boldsymbol{\alpha}}, -\mathbf{X}\tilde{\boldsymbol{\alpha}})$. Define the* reconstruction function *$R(\mathbf{z}, \xi) = R^{\alpha,\xi\to\theta}\left(R^{z\to\alpha}(\mathbf{z}), \xi\right)$. Consider a 2-layer neural net with ReLU, absolute value, or leaky ReLU activation. Given a Lasso solution $(\mathbf{z}^*, \xi)$, the reconstructed neural net parameters are $\theta = (\boldsymbol{\alpha}, \xi, \mathbf{w}, \mathbf{b}) = R(\mathbf{z}^*, \xi)$.*

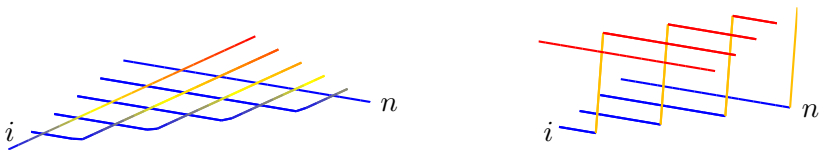


Figure 4: Generic shape of $\mathbf{A}_+ \in \mathbb{R}^{N\times N}$ defined by $\mathbf{A}_{+i,n} = \sigma(x_i - x_n)$, where σ is ReLU (left) and sign activation (right). Each i^{th} curve represents a feature. The points $\big(i, n, \mathbf{A}_{+i,n}\big)$ are plotted in 3-D, with $\mathbf{A}_{+i,n}$ represented by the curve height and color. Here, $n \in [N]$ but each curve interpolates between integer values of n.

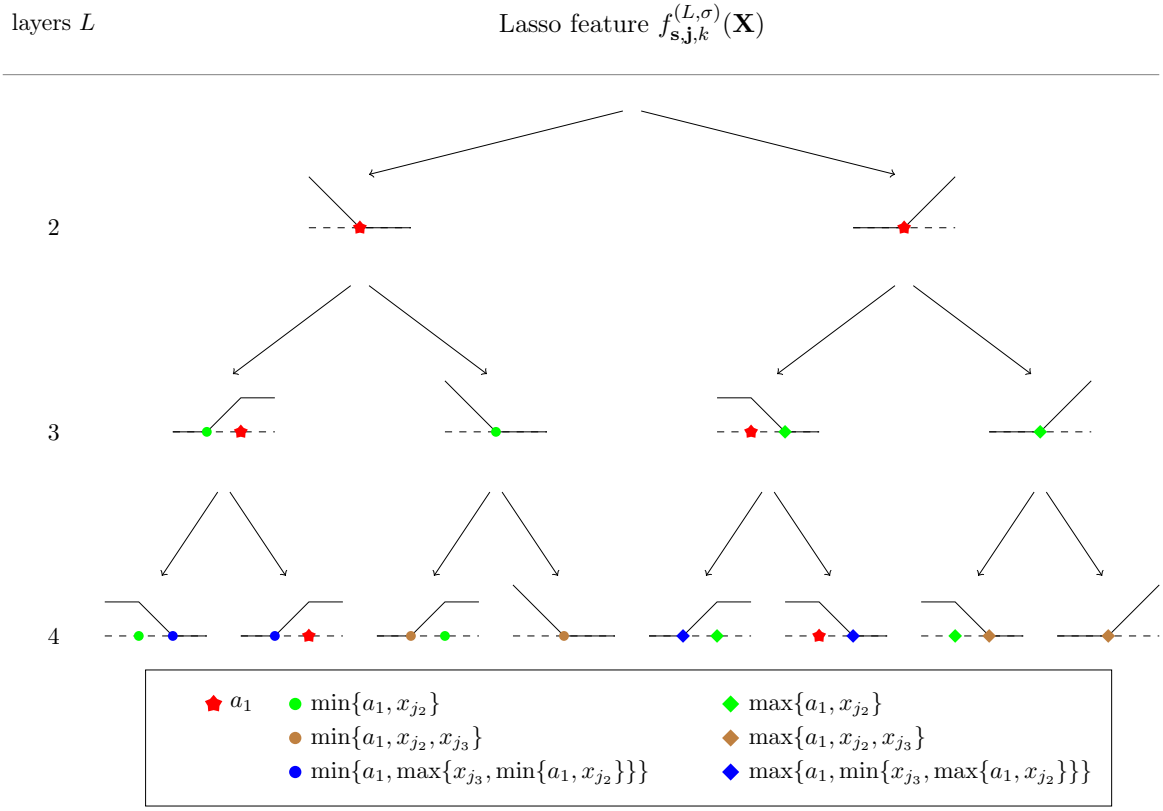


Figure 5: The L^{th} row consists of possible graphs of the dictionary function $f^{(L,\sigma)}_{\mathbf{s},\mathbf{j},k}(x)$ where σ is ReLU and $\mathbf{s}$ is varied. The point a_1 is as defined in Definition 4. Arrows point to the right and left to represent the cases $s_l \geq 0$ or $s_l \leq 0$, respectively, for $l \in [L-1]$.

3.2 Deep neural networks with sign activation

In this section, we analyze the training problem of an L-layer deep network with sign activation, which need not be a deep narrow network. We say the vector $\mathbf{h} \in \{-1, 1\}^N$ *switches* at $n > 1$ if $h_n \neq h_{n-1}$. For $n \in \mathbb{N}$, let the *switching set* $\mathbf{H}^{(n)}$ be the set of all vectors in $\{-1, 1\}^N$ that start with 1 and switch at most n times.

Lemma 2 *For $L = 2$ and sign activation, the Lasso dictionary in Theorem 1 is $\mathbf{H}^{(1)}$.*

The next result shows that the training problem (1) for deeper networks with sign activation is also equivalent to a Lasso problem (2) whose dictionary is a switching set. Proofs in this section are deferred to Appendix D.1.

Theorem 2 *Consider a Lasso problem whose dictionary is the switching set* $\mathbf{H}^{(K)}$*,* $\xi = 0$*, and with solution* $\mathbf{z}^*$*. Let* $m^* = \|\mathbf{z}^*\|_0$*. This Lasso problem is equivalent to the training problem for a neural network with sign activation,* $m_L \geq m^*$*, and* $m_{L-2} = K$ *when it is a rectangular network, and* $\prod_{l=1}^{L-1} m_l = K$ *when it is a tree network.*

Theorem 2 generalizes Theorem 1 for sign networks. By Lemma 2, for $L = 2$, $1 = d = m_0 = L - 2$ so the dictionary with sign activation is also $\mathbf{H}^{(m_{L-2})} = \mathbf{H}^{(1)}$. Adding another layer to a parallel network with sign activation expands the dictionary to vectors with up to m_1 switches. But adding even more layers doesn't change the dictionary, unless the neural net is a tree architecture: then, the features have as many breakpoints as the product of the number of neurons in each layer. The Lasso representation suggests that the representation power of networks with sign activation may stagnate after three layers. Moreover, the sign activation dictionary has no reflection features, which also may limit its expressability (Minsky and Papert, 2017). Reflection features allow neural networks to fit functions with breakpoints at locations in between data points. The reflection breakpoints for ReLU networks suggest that they can learn geometric structures or symmetries from the data. An explicit reconstruction of an optimal neural net with sign activation for $L = 3$ layers is described next. It is drawn in Figure 6. he reconstruction uses the unscaling defined in Section 2.

Lemma 3 *Consider a* $L = 3$*-layer sign-activated network. Suppose* $\mathbf{z}^*$ *is optimal in the Lasso problem, and* $m_L \geq ||\mathbf{z}^*||_0$*. Let* $\xi = 0$*. Let* $\boldsymbol{\alpha} = \mathbf{z}^*$*. Suppose* $\mathbf{A}_i$ *switches at* $I_1^{(i)} < I_2^{(i)} < \cdots < I_{m^{(i)}}^{(i)}$*. Let* $\mathbf{W}_n^{(i,1)} = 1, \mathbf{b}_n^{(i,1)} = -x_{I_n^{(i)}-1}, \mathbf{W}_n^{(i,2)} = (-1)^{n+1}$ *and* $\mathbf{b}^{(i,2)} = -\mathbf{1}\left\{m^{(i)} \text{ odd}\right\}$*. Let all amplitude parameters be* 1*. Let* $\mathcal{I} = \{i : z_i \neq 0\}$*. If* $i \notin \mathcal{I}$*, set* $s^{(i,l)}, \alpha_i, \mathbf{W}^{(i,l)}, \mathbf{b}^{(i,l)}$ *to zero. These parameters are optimal when unscaled.*

The reconstruction of a 3-layer network with sign activation is efficient and explicit. Reconstructions for other architectures are given in Appendix D.1. The Lasso dictionary for deep neural nets in previous work (Ergen et al., 2023) uses a dictionary that depends on the training data $\mathbf{X}$. However, Lemma 2 and Theorem 2 show that networks with sign activation have dictionaries that are invariant to the training data $\mathbf{X}$. So to train multiple neural nets on different data, the dictionary matrix $\mathbf{A}$ only needs to be constructed once. Using the Lasso problem, the next result compares the training loss for networks with sign activation for different depths.

Corollary 1 *Consider a sign-activated neural network with* $L = 3$ *layers. There exists an equivalent* $L = 2$*-layer network with* $m_{L=2} = m_1 m_3$ *neurons, where the* m_1, m_3 *are the number of neurons in the* 3*-layer network. Moreover, let* $p^*_{L,\beta}$ *be the optimal value of the training problem* (1) *for* L *layers, regularization* β *and sign activation. Let* m_1, m_2 *be the number of neurons in the first and second hidden layer of a three layer net, respectively. Then, for two-layer nets trained with at least* $m_1 m_2$ *hidden neurons,* $p^*_{L=3,\beta} \leq p^*_{L=2,\beta} \leq p^*_{L=3,m_1\beta}$*.*

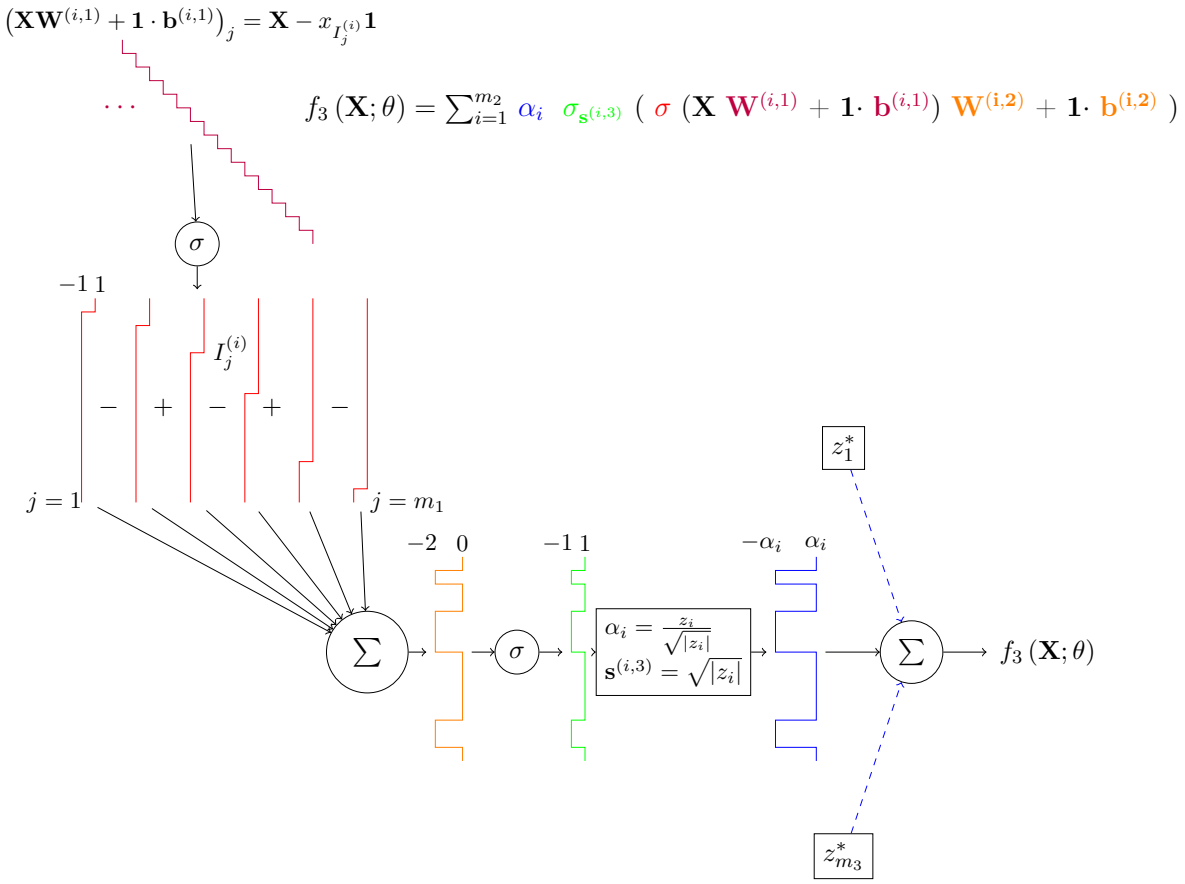


Figure 6: Output of an optimal 3-layer neural net with sign activation reconstructed from a Lasso solution $\mathbf{z}^*$ using Lemma 3 . The pulse colors correspond to network operations. The alternating $+, -$ represent $\mathbf{W}^{(i,2)} = (1, -1, 1, -1, \cdots)$. The red and green pulses illustrate 2 and 3-layer dictionary features, respectively (Theorem 1, Theorem 2), while the other colors represent multiplication by weights and amplitudes.

Corollary 1 states that a 3-layer net can achieve lower training loss than a 2-layer net, but only while its regularization β is at most m_1 times stronger. Analysis of the span or uniqueness and the generalizing abilities of different optimal or stationary solutions to (1) is an area for future work. Next, we give an extension of the Lasso equivalence for 2-D data.

3.2.1 Example of 2-D data

The next result extends Theorem 2 to 2-D data on the upper half plane. We consider parallel neural nets without internal bias parameters, that is, $\mathbf{b}^{(i,l)} = 0$ for all i, l. Proofs are located in Appendix D.2.

Theorem 3 *Consider a Lasso problem whose dictionary is the switching set* $\mathbf{H}^{(K)}$, $\xi = 0$, *and with solution* $\mathbf{z}^*$. *Let* $m^* = \|\mathbf{z}^*\|_0$. *This Lasso problem is equivalent to the training*

problem for a sign-activated network without internal biases that is 2-layer or rectangular, satisfies $m_L \geq m^*$, $m_{L-2} = K$, and is trained on 2-D data with unique angles in $(0, \pi)$.

The next result reconstructs an optimal neural net from the Lasso problem in Theorem 3.

Lemma 4 *Let $\mathbf{R}_{\frac{\pi}{2}} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$ be the counterclockwise rotation matrix by $\frac{\pi}{2}$. An optimal parameter set for the training problem in Theorem 3 when $L = 2$ is the unscaled version of $\theta = \left\{\alpha_i = z_i^*, \mathbf{s}^{(i,1)} = 1, \mathbf{W}^{(i,1)} = \mathbf{R}_{\frac{\pi}{2}} \left(\mathbf{x}^{(i)}\right)^T, \xi = 0 : z_i^* \neq 0\right\}$, where $\mathbf{z}^*$ is optimal in the Lasso problem.*

Remark 4 *A Lasso dictionary for an architecture discussed in Theorem 1, Theorem 2 or Theorem 3 is a superset of any dictionary with the same architecture but shallower depth.*

Reconstructing a neural net from a Lasso solution gives at least one optimal neural net in the non-convex training problem (1). The next section discusses the entire solution set to the Lasso problem, and how this generates a subset of optimal networks in (1).

4 The solution sets of Lasso and the training problem

We have shown that training neural networks on 1-D data is equivalent to fitting a Lasso model. Now we develop analytical expressions for all minima of the Lasso problem and its relationship to the set of all minima of the training problem. These results, which build on the existing literature for convex reformulations (Mishkin and Pilanci, 2023) as well as characterizations of the Lasso (Efron et al., 2004), illustrate that the Lasso model provides insight into non-convex networks. We focus on two-layer models with ReLU, leaky ReLU and absolute value activations, although our results can be extended to other architectures by considering the corresponding neural net reconstruction. Proofs are deferred to Appendix H.

We start by characterizing the set of global optima to the Lasso problem (2). Suppose $(\mathbf{z}^*, \xi^*)$ is a solution to the convex training problem. In this notation, the optimal model fit $\hat{\mathbf{y}}$ and equicorrelation set $\mathcal{E}_\beta$ are given by

$$\hat{\mathbf{y}} = \mathbf{A}\mathbf{z}^* + \xi^* \mathbf{1}, \qquad \mathcal{E}_\beta = \left\{ i : |\mathbf{A}_i^\top (\hat{\mathbf{y}} - \mathbf{y})| = \beta \right\},$$

where $\hat{\mathbf{y}}$ is unique over the optimal set (Vaiter et al., 2012; Tibshirani, 2013). The equicorrelation set contains the features maximally correlated with the residual $\hat{\mathbf{y}} - \mathbf{y}$ and plays a critical role in the solution set.

Proposition 1 *Suppose $\beta > 0$. Then the set of global optima of the Lasso problem (2) is*

$$\Phi^*(\beta) = \left\{ (\mathbf{z}, \xi) : z_i \neq 0 \Rightarrow sign(z_i) = sign\left(\mathbf{A}_i^\top (\hat{\mathbf{y}} - \mathbf{y})\right),\ z_i = 0 \,\forall i \notin \mathcal{E}_\beta,\ \mathbf{A}\mathbf{z} + \xi \mathbf{1} = \hat{\mathbf{y}}. \right\} \quad (6)$$

The solution set $\Phi^*(\beta)$ is polyhedral and its vertices correspond exactly to minimal models, i.e. models with the fewest non-zero elements of $\mathbf{z}$ (Mishkin and Pilanci, 2023). Let R be the reconstruction function described in Remark 3. All networks generated from applying R to a Lasso solution are globally optimal in the training problem. The next result gives a full description of such networks. The 2-layer parameter notation defined in Definition 7 is used.

Proposition 2 *Suppose $\beta > 0$ and the activation is ReLU, leaky ReLU or absolute value. The set of all 2-layer Lasso-reconstructed networks is*

$$R(\Phi(\beta)) = \Bigg\{ (\mathbf{w}, \mathbf{b}, \alpha, \xi) : \alpha_i \neq 0 \Rightarrow sign(\alpha_i) = sign\left(\mathbf{A}_i^\top (\mathbf{y} - \hat{\mathbf{y}})\right), b_i = -x_i \frac{\tilde{\alpha}_i}{\sqrt{|\alpha_i|}}, \\ w_i = \frac{\tilde{\alpha}_i}{\sqrt{|\alpha_i|}};\ \alpha_i = 0 \,\forall i \notin \mathcal{E}_\beta,\ f_2(\mathbf{X}; \theta) = \hat{\mathbf{y}} \Bigg\}. \tag{7}$$

Proposition 2 shows that all neural nets trained using our Lasso and reconstruction share the same model fit whose set of active neurons is at most the equicorrelation set. By finding just *one* optimal neural net that solves Lasso, we can form $R(\Phi^(\beta))$ and compute all others.

The min-norm solution path is continuous for the Lasso problem (Tibshirani, 2013). Since the solution mapping in Definition 6, Appendix C is continuous, the corresponding reconstructed neural net path is also continuous as long as the network is sufficiently wide. Moreover, we can compute this path efficiently using the LARS algorithm (Efron et al., 2004). This is in contrast to the under-parameterized setting, where the regularization path is discontinuous (Mishkin and Pilanci, 2023).

What subset of optimal, or more generally, stationary, points of the non-convex training problem (1) consist of Lasso-generated networks $R(\Phi(\beta))$? First, $R(\Phi(\beta))$ can generate additional optimal networks through neuron splitting, described as follows. Consider a single neuron $\alpha\sigma_s(wx + b)$ (where $\alpha, w, b \in \mathbb{R}$), and let $\{\gamma_i\}_{i=1}^n \subset [0,1]^n$ be such that $\sum_{i=1}^n \gamma_i = 1$. The neuron can be *split* into n neurons $\left\{\sqrt{\gamma_i}\alpha\sigma\left(\sqrt{\gamma_i}wx + \sqrt{\gamma_i}b\right)\right\}_{i=1}^n$ Wang et al. (2021). For any collection Θ of parameter sets θ, let $P(\Theta)$ be the collection of parameter sets generated by all possible neuron splits and permutations of each $\theta \in \Theta$. Next, let $\mathcal{C}(\beta)$ and $\tilde{C}(\beta)$ be the sets of Clarke stationary points and solutions to the non-convex training problem (1), respectively.

Proposition 3 *Suppose $L = 2, \beta > 0$, the activation is ReLU, leaky ReLU or absolute value and $m^* \leq m \leq |\mathcal{M}| = 2N$. Let $\Theta^P = \{\theta : \forall i \in [m], \exists j \in [N] \text{ s.t. } b_i = -x_j w_i\}$. Then*

$$P(R(\Phi(\beta))) = \tilde{\mathcal{C}}(\beta) \cap \Theta^P = \mathcal{C}(\beta) \cap \Theta^P. \tag{8}$$

Proposition 3 states that up to neuron splitting and permutation, our Lasso method gives all stationary points in the training problem satisfying $b_i = -x_i w_i$. Moreover, all such points are optimal in the training problem, similar to Feizi et al. (2017).

Since optimal solutions are stationary, a neural net reconstructed from the Lasso model is in $\tilde{C}(\beta) \subset \mathcal{C}(\beta)$. However, $C(\beta) \not\subset \Theta^P$. This is because there may be other neural nets with the same output on $\mathbf{X}$ as the reconstructed net so that they are all in $\mathcal{C}(\beta)$, but that differ in the the unregularized parameters $\mathbf{b}$ and ξ, so that they are not in Θ^P. For example, if β is large enough, the Lasso solution is $\mathbf{z} = \mathbf{0}$ (Efron et al., 2004), so the reconstructed net will have $\boldsymbol{\alpha} = \mathbf{0}$, which makes the neural net invariant to $\mathbf{b}$. In this section, we analyzed the general structure of the Lasso solution set when $\beta > 0$. Next, we analyze the Lasso solution set for specific activations and training data when $\beta \to 0$.

5 Solution sets of Lasso under minimal regularization

One of the insights that the Lasso formulation provides is that under minimal regularization, certain neural nets perfectly interpolate the data.

Corollary 2 *For the ReLU, absolute value, sign, and threshold networks with $L = 2$ layers, and sign-activated deeper networks, if $m_L \geq m^*$, then $f_L(\mathbf{X}; \theta) \to \mathbf{y}$ as $\beta \to 0$.*

Proofs in this section are deferred to Appendix E. In Corollary 2, m^* depends on L and the activation and is defined in Theorem 1 and Theorem 2. The Lasso equivalence and reconstruction also shed light on optimal neural network structure as regularization decreases. The *minimum (l_1) norm subject to interpolation* version of the Lasso problem is

$$\min_{\mathbf{z},\xi} \|\mathbf{z}\|_1, \text{ s.t. } \mathbf{A}\mathbf{z} + \xi\mathbf{1} = \mathbf{y}. \tag{9}$$

Loosely speaking, as $\beta \to 0$, if $\mathbf{A}$ has full column rank, the Lasso problem (2) "approaches" the minimum norm problem (9), where $\xi = 0$ for sign and threshold activations. The rest of this section describes the solution sets of (9) for certain networks.

Proposition 4 *Let $L = 2$. Suppose σ is the absolute value activation. Let $\mathbf{z}^*$ be a solution to (9). Then, we have $z_1^* z_n^* \leq 0$. Moreover, the entire solution set of (9) for $\mathbf{z}^*$ is given by*

$$\left\{ \mathbf{z}^* + t \text{ sign}(z_1^*)(1, 0, \cdots, 0, 1)^T \middle| -|z_1^*| \leq t \leq |z_N^*| \right\}. \tag{10}$$

Proposition 5 *Let $L = 2$. Suppose σ is sign activation. Then, for $\beta \geq 0$, the Lasso problem (2) has a unique solution. And the minimum norm solution $\mathbf{z}^*$ to (9) is $\mathbf{z}^* = \mathbf{A}^{-1}\mathbf{y}$.*

Given an optimal bias term ξ^*, if $\mathbf{A}$ is invertible, then $\mathbf{z}^* = \mathbf{A}^{-1}(\mathbf{y} - \xi^*\mathbf{1})$ is optimal in (9). Appendix F finds $\mathbf{A}^{-1}$ for some activation functions. The structure of $\mathbf{A}^{-1}$ suggests the behavior of neural networks under minimal regularization: sign-activated neural networks act as difference detectors, while neural networks with absolute value activation, whose subgradient is the sign activation, act as a second-order difference detectors (see Remark 20). The next result shows that threshold-activated neural networks are also difference detectors, but for the special case of positive, nonincreasing y_n. An example of such data is cumulative revenue, e.g. $y_n = \sum_{i=1}^n r_i$ where r_i is the revenue in dollars earned on day i.

Proposition 6 *Let $L = 2$. Suppose σ is threshold activation and $y_1 \geq \cdots \geq y_N \geq 0$. Then*

$$z_n^* = \begin{cases} y_n - y_{n-1} & \text{if } n \leq N-1 \\ y_N & \text{if } n = N \\ 0 & \text{else} \end{cases}$$

is the unique solution to the minimum norm problem (9).

The next result gives a lower bound on the optimal value of the minimum weight problem for ReLU networks. If we can find $\mathbf{z}$ with a l_1 norm that meets the lower bound and a ξ such that $\mathbf{A}\mathbf{z} + \xi\mathbf{1} = \mathbf{y}$, then we know $\mathbf{z}, \xi$ is optimal. In this section, for $n \in [N-1]$, let $\mu_n = \frac{y_n - y_{n+1}}{x_n - x_{n+1}}$ be the slope between the n^{th} and $n + 1^{th}$ data points. Let $\mu_N = 0$.

Lemma 5 *The optimal value $\|\mathbf{z}^*\|_1$ of the minimum norm problem (9) for $L = 2, 3, 4$ and ReLU activation is at least $\max_{n \in [N-1]} |\mu_n|$.*

In the special case of 2-layer networks, the next result gives a solution to the minimum weight problem. For $i \in [N]$, let $(z_+)_i$ and $(z_-)_i$ be the Lasso variable corresponding to the features $\text{ReLU}^+_{x_i}$ and $\text{ReLU}^-_{x_i}$, respectively. In other words, $\mathbf{z}_+$ corresponds to $\mathbf{A}_+$ and $\mathbf{z}_-$ corresponds to $\mathbf{A}_-$ as defined in Remark 2.

Lemma 6 *The optimal value for the minimum norm problem* (9) *for* $L = 2$ *and ReLU activation is* $\|\mathbf{z}^*\|_1 = \sum_{n=1}^{N-1} |\mu_n - \mu_{n+1}|$. *An optimal solution is* $(z_+)_{n+1} = \mu_n - \mu_{n+1}$ *for* $n \in [N-1]$, $\mathbf{z}_- = \mathbf{0}$, *and* $\xi = y_N$.

We examined neural networks when $\beta \to 0$. We next analyze networks as β grows.

6 Solution path for sign activation and binary, periodic labels

This section examines solution paths of Lasso problems for 2 and 3-layer neural nets with sign activation and 1-D data where $\mathbf{y}$ is binary. Such data appears in temporal sequences such as binary encodings of messages communicated digitally, (Kim et al., 2018), neuron firings in the brain (Fang et al., 2010), and other applications, where x_n represents time. These real world sequences are in general aperiodic. However, in the special case that the target vector is periodic and binary, the Lasso problem gives tractable solutions for optimal neural networks. This offers a step towards analyzing neural network behavior for more general, aperiodic data, which is an area for future work. We call the binary, periodic sequence a square wave, defined as follows. For a positive even integer T that divides N, define a *square wave* to be $\mathbf{h}^{(T)} \in \{-1, 1\}^N$ that starts with 1 and is periodic with period T. Given a square wave of period T, let $k = \frac{N}{T}$ be the number of cycles it has. If the real line is split into a finite number of regions by binary labels, the square wave represents the labels of a monotone sequence of points, with the same number of samples in each region. The right-hand graph of Figure 15 (Appendix G) plots the elements of a square wave over its indices.

There is a *critical value* $\beta_c = \max_{n \in [N]} |\mathbf{A}_n^T \mathbf{y}|$ such that when $\beta > \beta_c$, the solution of the Lasso problem has $\mathbf{z}^*$ as the all-zero vector (Efron et al., 2004). Let $\tilde{\beta} = \frac{\beta}{\beta_c}$. Theorem 1 specifies the Lasso problem for a 2-layer network with sign activation. We will use the $N \times N$ dictionary matrix $\mathbf{A}$ with $\mathbf{A}_{i,n} = \sigma(x_i - x_n)$, as defined in Remark 2. The Lasso solution $\mathbf{z}^* \in \mathbb{R}^N$ is unique, by Proposition 5. The next results gives the entire solution path of this Lasso problem and an optimal neural net in closed form for a square wave target vector. Proofs in this section are deferred to Appendix G.

Theorem 4 *Consider the Lasso problem for a* 2*-layer net with sign activation and square wave target vector of period* T. *The critical value is* $\beta_c = T$. *And the solution is*

$$z^*_{\frac{T}{2}i} = \begin{cases} \begin{cases} \frac{1}{2}\left(1 - \tilde{\beta}\right)_+ & \text{if } i \in \{1, 2k-1\} \\ 0 & \text{else} \end{cases} & \text{if } \tilde{\beta} \geq \frac{1}{2} \\ \begin{cases} 1 - \frac{3}{2}\tilde{\beta} & \text{if } i \in \{1, 2k-1\} \\ (-1)^{i+1}\left(1 - 2\tilde{\beta}\right) & \text{else} \end{cases} & \text{if } \tilde{\beta} \leq \frac{1}{2}. \end{cases}, \tag{11}$$

for $i \in [2k-1]$ *and* $z^*_n = 0$ *at all other* $n \in [N]$.

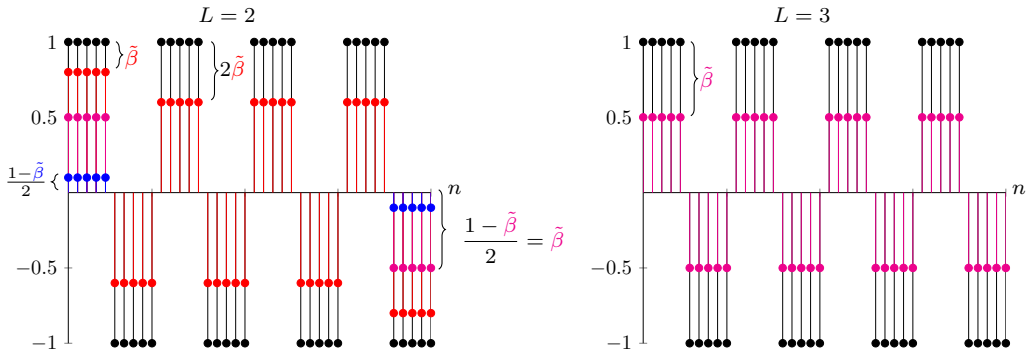


Figure 7: Each of the two figures depicts (n, y_n) with black dots, where $\mathbf{y} = \mathbf{h}^{(T)}$ with $T = 10, N = 40$. Sign-activated neural net predictions are depicted as $(n, f_L(x_n; \theta))$ with blue, magenta, and red dots for $\tilde{\beta} = \frac{4}{5} \in \left[\frac{1}{2}, 1\right]$, $\tilde{\beta} = \frac{1}{2}$, and $\tilde{\beta} = \frac{1}{5} \leq \frac{1}{2}$, respectively.

Corollary 3 *For a square wave target vector with period T, there is an optimal 2-layer neural network with sign activation specified by*

$$f_2(x;\theta) = 0, \qquad \text{if } \tilde{\beta} \geq 1$$

$$f_2(x;\theta) = \begin{cases} -\left(1-\tilde{\beta}\right) & \text{if } x < x_{N-\frac{T}{2}} \\ 0 & \text{if } x_{N-\frac{T}{2}} \leq x < x_{\frac{T}{2}} \\ 1-\tilde{\beta} & \text{if } x \geq x_{\frac{T}{2}} \end{cases} \qquad \text{if } \frac{1}{2} \leq \tilde{\beta} \leq 1$$

$$f_2(x;\theta) = \begin{cases} -\left(1-\tilde{\beta}\right) & \text{if } x < x_{N-\frac{T}{2}} \\ (-1)^i\left(1-2\tilde{\beta}\right) & \text{if } x_{\frac{T}{2}(i+1)} \leq x < x_{\frac{T}{2}i}, \quad i \in [2k-2] \\ 1-\tilde{\beta} & \text{if } x \geq x_{\frac{T}{2}} \end{cases} \qquad \text{if } \tilde{\beta} \leq \frac{1}{2}$$

Theorem 4 implies that when $\beta > T$, an optimal neural net is the constant zero function. In Corollary 3, when $\beta \leq \frac{T}{2}$, $f_2(\mathbf{X};\theta)$ is periodic over $\left[\frac{T}{2}, N - \frac{T}{2}\right]$ with period T, and has amplitude $2\frac{\beta}{T}$ less than that of $\mathbf{y}$. The next results give the solution path and an optimal neural net when $L = 3$, and are proved in Appendix G.2.

Theorem 5 *Consider the Lasso problem for a 3-layer network with sign activation and target vector a square wave of period T and $m_3 \geq 2\frac{T}{N} - 1$. Then $\beta_c = N$ and $\mathbf{A}_i = -\mathbf{h}^{(T)}$ for some i. The solution to the Lasso problem is $z_i^* = -\left(1-\tilde{\beta}\right)_+$ and $z_n^* = 0$ at all other n.*

Corollary 4 *Let $x_0 = \infty$. For a square wave target vector with period T, there is an optimal 3-layer neural net with sign activation specified by $f_3(x;\theta) = (1-\tilde{\beta})_+(-1)^{(i-1)}$ if $x_{\frac{T}{2}i} \leq x < x_{\frac{T}{2}(i-1)}$ for $i \in [2k-1]$, and $f_3(x;\theta) = -(1-\tilde{\beta})_+$ if $x < x_{N-\frac{T}{2}}$.*

Since only one parallel unit is active in this network, it is also a standard neural net. The neural net has output $f_3(x;\theta)(\mathbf{X}) = (1-\tilde{\beta})_+\mathbf{y}$. If $\beta > N$, then the optimal neural net is the constant zero function.

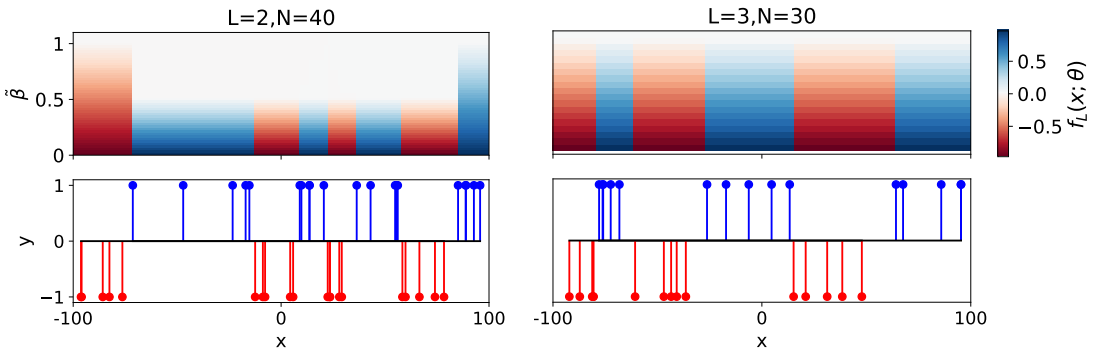


Figure 8: The bottom figures plot training data (x_n, y_n). The top figures plot sign-activated neural net predictions by color for each x, as parameterized by β on the vertical axis.

Figure 7 illustrates $f_L(\mathbf{X};\theta)$ when $\mathbf{y} = \mathbf{h}^{(T)}$. Consider the 2-layer network in the left plot. When $\beta \to 0$, the network interpolates the target vector perfectly. As $\tilde{\beta}$ increases to $\frac{1}{2}$ (red dots) from 0, the magnitude of the middle segments decrease at a faster rate than the outer segments until at $\tilde{\beta} = \frac{1}{2}$, the net consists of just the outer segments (magenta dots). As $\tilde{\beta}$ increases to 1 from $\frac{1}{2}$ (blue dots), these outer segments decrease until the neural net is the zero function. The solution path suggests that as the regularization increases, the 2-layer network focuses on preserving the boundary points of the data (first and last T points) to be close to the target vector. Therefore the network will generalize well if noise occurs in the middle of the data. In contrast, if noise occurs uniformly over the data, the 3-layer network will generalize well.

We verify Theorem 4 and Theorem 5 by solving the Lasso problem on training data that is chosen from a uniform distribution on $[-100, 100]$ and target vector $\mathbf{h}^{(T)}$. Figure 8 illustrates the training data and neural net predictions. Suppose we use the neural net as a binary classifier whose output is the sign of $f_L(x;\theta)$, where the network is "undecided" if $f_L(x;\theta) = 0$.The red, blue and white indicate classifications of -1, 1, and "undecided," respectively. For $\beta < \beta_c$, the 3-layer net always classifies the training data accurately, but the 2-layer net is undecided on all but the first and last interval if $\beta > \beta_c/2$. When used as a regressor, for each β, the magnitude of the 3-layer net's prediction is the same over all samples, while the 2-layer net is biased toward a stronger prediction on the first and last intervals. In this sense, the 3-layer network generalizes better. In addition, the 3-layer net changes more uniformly with β than the 2-layer net, making it easier to tune β. In this example with a square wave target vector, we analytically solve the neural net Lasso problem for $\beta \in (0, \infty)$ and verify our results by numerically solving the Lasso problem. This gives analytical expressions for optimal neural nets. In the next section, we analyze other examples of training data, analytically solve the min norm version of the Lasso problem and verify our results experimentally by training neural nets with the non-convex problem.

7 Numerical results

Here we describe simulations that support our theoretical results. In Figure 2, we compare neural nets trained using the non-convex and convex Lasso problems given in (1) and (2). In

order to find a near-optimal solution to (2), we first find analytical optimal solutions $(\mathbf{z}^*, \xi^*)$ to the minimum norm problem (9). The nonzero components z_i^* and their corresponding Lasso features specify their solutions $\mathbf{z}^*$ and are shown in the third column of Figure 2. Our optimal solutions satisfy $\xi^* = 0$. Our solutions are optimal by Lemma 6 for $L = 2$ and Lemma 5 for $L > 2$. Then, we numerically solve the training problem for a standard network. We use a β sufficiently small such that if p^* and $\hat{p}^*$ are the optimal values respectively found by analytically solving the min norm problem (9), which has objective $\|\mathbf{z}\|_1$, and numerically solving the Lasso problem (2), which has objective $\frac{1}{2}\|\mathbf{A}\mathbf{z} + \xi\mathbf{1} - \mathbf{y}\|_2^2 + \beta\|\mathbf{z}\|_1$, then $\left|p^* - \frac{\hat{p}^*}{\beta}\right| < 10^{-3}$; in other words the optimal Lasso objective is approaching $\beta\|\mathbf{z}^*\|_1$. We set $\beta = 10^{-7}$ to satisfy this requirement. We use a standard network in the non-convex model to show that our Lasso formulation is applicable even to standard architectures. To optimize the training problem, we use Adam with a learning rate of $5(10^{-3})$ and weight decay of 10^{-4}. Using SGD appeared to give similar results as Adam. The number of final-layer neurons m_L and epochs is 100 and 10^3 for $L = 2$; 500 and 10^5 for $L = 3$; and 100 and $5(10^4)$ for $L = 4$.

The capped ramp features allow the $L = 3$ neural net to achieve a lower objective of $\|\mathbf{z}^*\|_1 = 1$ in the min norm problem compared to the $L = 2$ net, which achieves an objective value of $\|\mathbf{z}^*\|_1 = 2$. The optimal value of the min norm problem for $L = 4$ is also $\|\mathbf{z}^*\|_1 = 1$. Therefore the solution shown in Figure 2 for $L = 3$ is also optimal for $L = 4$. The neural nets found from the non-convex training problem closely match those trained with Lasso. The neural nets trained with Adam has slightly suboptimal fit to the data compared to the Lasso min norm solution, but this is likely due to finite training time, solver tolerance, the solver computing a near-globally optimal solution, and the Lasso min norm solution being an approximation to the Lasso problem. As seen in Figure 2, $L \in \{2, 3\}$, the breakpoints in the Lasso and non-convex neural nets occur only at the training data points. But when $L = 4$, the Lasso and non-convex neural nets have a breakpoint at $x = 8$, which is not a data point but a **reflection** $R_{(0,4)}$ of data points. Appendix I.1 shows that reflection breakpoints can also appear in near-optimal solutions.

In addition to training ReLU networks under minimal β, we train neural networks with threshold activations and larger β. This experiment supports the usefulness of the Lasso problem for training neural nets. We generate 1-D data samples from an i.i.d. distribution $x \in \mathcal{N}(0, 1)$, and then label them with a Bernoulli random variable. We use $N = 40$ samples and $\beta = 10^{-3}$ to train a 2-layer neural network with threshold activation using the Lasso problem (2) as well as a non-convex training approach based on the Straight Through Estimator (STE) (Bengio et al., 2013). As illustrated in Figure 11, the convex training approach achieves significantly lower objective value than all of the non-convex trials with different seeds for initialization. Figure 11 also plots the predictions of the models. We observe that the non-convex training approach fits the data samples exactly on certain intervals but provides a poor overall function fitting, whereas our convex models yields a more reasonable piecewise linear fit. In particular, the neural net trained with the non-convex problem fits the data in Figure 11 poorly compared to Figure 2. This may occur because in Figure 11, the data set is larger and more complex, and STE training is used because of the threshold activation.

Next we present additional numerical results by applying our theory to real-world data.

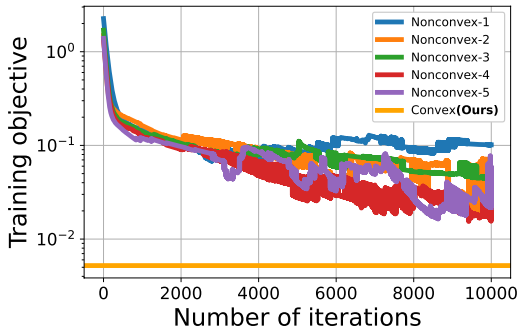


Figure 9: Training objective

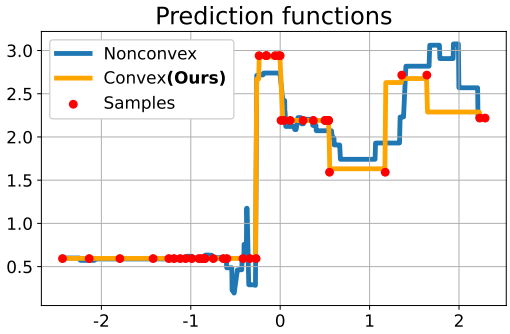


Figure 10: Function fit

Figure 11: Training objective (left) and function fit (right) for a neural net using the convex Lasso problem versus the non-convex training problem using STE.

8 Application: Time-series modeling

In this section, we apply the Lasso problem for neural networks to an autoregression problem. Suppose at times $1, \cdots, T+1$ we observe data points $x_1, \cdots, x_{T+1} \in \mathbb{R}$ that follow the time-series model

$$x_t = f(x_{t-1}; \theta) + \epsilon_t, \tag{12}$$

where $f : \mathbb{R} \to \mathbb{R}$ is parameterized by some parameter θ and $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$ represents observation noise. The parameter θ is unknown, and the goal is find θ that best fits the model (12) to the data $x_1, \cdots, x_{T+1}$. For example, the *auto-regressive model with lag* 1 (AR(1)) is a linear model $f(x; \theta) = ax$ where $\theta = a$ is chosen as a solution to $\min_{\theta \in \Theta} \sum_{t=1}^{T} (f(x_t; \theta) - x_{t+1})^2$. For a more expressive model, instead of $f(x; \theta) = ax$ suppose we use a 2-layer neural network

$$f_2^{\text{NN}}(x; \theta) = \sum_{i=1}^{m} |xw_i + b_i| \alpha_i, \tag{13}$$

which has m neurons and absolute value activation. The parameter set is $\theta = \{w_i, b_i, \alpha_i\}_{i=1}^m$. Suppose we choose θ that solves the *neural net (NN) autoregression training problem*

$$\min_{\theta \in \Theta} \frac{1}{T} \sum_{t=1}^{T} \left(f_2^{\text{NN}}(x_t; \theta) - x_{t+1} \right)^2 + \frac{\beta}{2} \|\theta_w\|_2^2. \tag{14}$$

By Theorem 1, this non-convex problem is equivalent to the convex Lasso problem (2) where $\mathbf{A}_{i,j} = |x_i - x_j|$ and $y_i = x_{i+1}$. Our models so far represent predictors of x_{t+1} from x_t. We can also find a neural network model $f_2^{\text{NN}}(x_t; \theta)$ (13) that represents the τ-quantile of the distribution of x_{t+1} given the observation x_t, where $\tau \in [0, 1]$, by using the *quantile regression loss* $L_\tau(z) = 0.5|z| + (\tau - 0.5)z$ and choosing θ that solves the *neural net (NN) quantile regression (QR) training problem*

$$\min_{\theta \in \Theta} \frac{1}{T} \sum_{t=1}^{T} L_\tau \left(f_2^{\text{NN}}(x_t; \theta) - x_{t+1} \right) + \frac{\beta}{2} \|\theta_w\|_2^2. \tag{15}$$

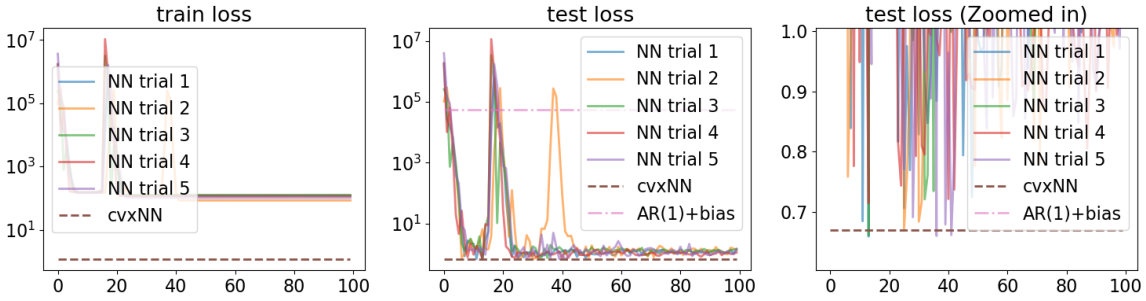


Figure 12: Comparison of neural autoregressive models of the form $x_t = f(x_{t-1};\theta) + \epsilon_t$ using convex and non-convex optimizers and the classical linear model AR(1) for time series forecasting. The horizontal axis is the training epoch. The dataset is BTC-2017min from Kaggle, which contains all 1-minute Bitcoin prices in 2017 (kag). The non-linear models outperform the linear AR(1) model. Moreover, SGD underperforms in training and test loss compared to the convex model which is guaranteed to find a global optimum of the NN objective.

Problem (15) can also be solved by converting it to an equivalent Lasso problem. We now compare solving the autoregression (14) and quantile regression (15) problems directly with 5 trials of stochastic gradient descent (SGD) initializations versus using the Lasso problem (2). We also compare against the baseline linear method $f(x;\theta) = ax + b$ (AR1+bias), where we include an additional bias term b.

We first build a neural network $f_2^{\mathrm{NN}}(x;\theta)$ (13) with m known, or *planted* neurons. We use this neural network to generate training samples $x_1, \ldots, x_{T+1}$ based on (12) with $f(x;\theta) = f_2^{\mathrm{NN}}(x;\theta)$ where $x_1 \sim \mathcal{N}(0,\sigma^2)$. Using the same model $f_2^{\mathrm{NN}}(x;\theta)$, we also generate test samples $x_1^{\mathrm{test}}, \ldots, x_{T+1}^{\mathrm{test}}$ in an analogous way. We use $T = 1000$ time samples. Then, we try to recover the planted neurons based on only the training samples by solving the NN AR/QR training problems.

In Figure 19, we present experiments based on the selection of m planted neurons and noise level σ^2. More results can be found in Appendix I.2. The neural net trained with Lasso is labeled cvxNN, which we observe has lower training loss. This appears to occur because different trials of NN (neural net trained directly without Lasso) get stuck into local minima. The global optimum that cvxNN reaches also enjoys effective generalization properties, as seen by the test loss. The *regularization path* is the optimal neural net's performance loss as a function of the regularization coefficient β. Figure 20 plots the regularization path for $\sigma^2 = 1$ and $m = 5$. The regularization path taken by cvxNN is smoother than NN, and can therefore be found more precisely and robustly by using the Lasso problem.

We also test upon real financial data for bitcoin price, including minutely bitcoin (BTC) price (BTC-2017min) and hourly BTC price (BTC-hourly). We consider the training problem on τ-quantile regression with $\tau = 0.3$ and $\tau = 0.7$. For each dataset, we first choose T data points as a training set and the consecutive T data points as a test set. The numerical results are presented in Figure 12. We observe that cvxNN provides a consistent lower bound on the training loss and demonstrates strong generalization properties, compared to large fluctuation in the loss curves of NN. More results can be found in Appendix I.2.

9 Conclusion

Our results show that deep neural networks with various activation functions trained on 1-D data with weight regularization can be recast as convex Lasso models with simple dictionary matrices. This provides critical insight into their solution path as the weight regularization changes. The Lasso problem also provides a fast way to train neural networks for 1-D data. Moreover, the understanding of the neural networks through Lasso models could also be used to explore designing better neural network architectures.

We proved that reflection features emerge in the Lasso dictionary whenever the depth is 4 or deeper. This leads to predictions that have breakpoints at reflections of data points about other data points. In contrast, for networks of depth 2 and 3, the breakpoints are located at a subset of training data. We believe that this mechanism enables deep neural networks to generalize to the unseen by encoding a geometric regularity prior.

The 1-D results can extend to sufficiently structured or low rank data in higher dimensions. Generalizing to higher dimensions is an area of future work. Building on a similar theme, (Pilanci, 2023) showed that the structure of hidden neurons can be expressed through convex optimization and Clifford's Geometric Algebra. The techniques developed in this paper can be combined with the Clifford Algebra to develop higher-dimensional analogues of the results.

10 Acknowledgements

This work was supported in part by the National Science Foundation (NSF) under Grant DMS-2134248; in part by the NSF CAREER Award under Grant CCF-2236829; in part by the U.S. Office of Naval Research (ONR) under Grant N00014-24-1-2164; in part by the U.S. Army Research Office Early Career Award under Grant W911NF-21-1-0242; in part by the Precourt Institute for Energy and the SystemX Alliance at Stanford University; in part by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1656518.

References

Kaggle. URL `www.kaggle.com`.

F. Bach. Breaking the curse of dimensionality with convex neural networks. *JMLR*, 18(1): 629–681, 2017.

Y. Bengio, N. Roux, P. Vincent, O. Delalleau, and P. Marcotte. Convex neural networks. In *Advances in Neural Information Processing Systems*, volume 18. MIT Press, 2005.

Y. Bengio, N. Léonard, and A. C. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *ArXiv:1308.3432*, 2013.

M. J. Bianco, P. Gerstoft, J. Traer, E. Ozanich, M. A. Roch, S. Gannot, and C.-A. Deledalle. Machine learning in acoustics: Theory and applications. *The Journal of the Acoustical Society of America*, 146(5):3590–3628, 2019.

J. M. Borwein and A. S. Lewis. *Convex Analysis and Nonlinear Optimization: Theory and Examples*. Springer, 2000.

S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

A. Bulat and G. Tzimiropoulos. XNOR-Net++: Improved binary neural networks. *ArXiv*, abs/1909.13863, 2019.

P. Chen and O. Ghattas. Projected Stein variational gradient descent. *Advances in Neural Information Processing Systems*, 33:1947–1958, 2020.

P. Chen, K. Wu, J. Chen, T. O'Leary-Roseberry, and O. Ghattas. Projected Stein variational newton: A fast and scalable Bayesian inference method in high dimensions. *Advances in Neural Information Processing Systems*, 32, 2019.

T. M. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, (3): 326–334, 1965.

B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

T. Ergen and M. Pilanci. Convex geometry of two-layer ReLU networks: Implicit autoencoding and interpretable models. PMLR, pages 4024–4033, 26–28 Aug. 2020.

T. Ergen and M. Pilanci. Convex geometry and duality of over-parameterized neural networks. *The Journal of Machine Learning Research*, 22(1):9646–9708, 2021a.

T. Ergen and M. Pilanci. Revealing the structure of deep neural networks via convex duality. In *ICML*, pages 3004–3014. PMLR, 2021b.

T. Ergen, H. I. Gulluk, J. Lacotte, and M. Pilanci. Globally optimal training of neural networks with threshold activation functions. *arXiv:2303.03382*, 2023.

C. Fang, Y. Gu, W. Zhang, and T. Zhang. Convex formulation of overparameterized deep neural networks. *arXiv:1911.07626*, 2019.

H. Fang, Y. Wang, and J. He. Spiking neural networks for cortical neuronal spike train decoding. *Neural Computation*, 22(4):1060–1085, 2010.

S. Feizi, H. Javadi, J. M. Zhang, and D. Tse. Porcupine neural networks: (almost) all local optima are global. *ArXiv*, abs/1710.02196, 2017.

M. Freitag, S. Amiriparian, S. Pugachevskiy, N. Cummins, and B. Schuller. audeep: Unsupervised learning of representations from audio with deep recurrent neural networks. *JMLR*, 18(1):6340–6344, 2017.

C.-L. Hsu and J.-S. R. Jang. On the improvement of singing voice separation for monaural recordings using the MIR-1K dataset. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(2):310–319, 2009.

N. Joshi, G. Vardi, and N. Srebro. Noisy interpolation learning with shallow univariate relu networks. *arXiv.2307.15396*, 07 2023.

K. Karhadkar, M. Murray, H. Tseran, and G. Montúfar. Mildly overparameterized relu networks have a favorable loss landscape. *arXiv:2305.19510*, 2023.

H. Kim, Y. Jiang, R. Rana, S. Kannan, S. Oh, and P. Viswanath. Communication algorithms via deep learning. *arXiv:1805.09317*, 2018.

M. Kim and P. Smaragdis. Bitwise neural networks for efficient single-channel source separation. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 701–705, 2018.

G. Kornowski, G. Yehudai, and O. Shamir. From tempered to benign overfitting in ReLU neural networks. *arXiv:2305.15141*, 2023.

S. Mavaddati. A novel singing voice separation method based on a learnable decomposition technique. *Circuits, Systems, and Signal Processing*, 39(7):3652–3681, 2020.

M. Minsky and S. A. Papert. *Perceptrons: An Introduction to Computational Geometry*. The MIT Press, 09 2017. ISBN 9780262343930. doi: 10.7551/mitpress/11301.001.0001. URL `https://doi.org/10.7551/mitpress/11301.001.0001`.

A. Mishkin and M. Pilanci. Optimal sets and solution paths of ReLU networks. In *International Conference on Machine Learning, ICML 2023*. PMLR, 2023.

M. Pilanci. From complexity to clarity: Analytical expressions of deep neural network weights via Clifford's geometric algebra and convexity. *arXiv:2309.16512*, 2023.

M. Pilanci and T. Ergen. Neural networks are convex regularizers: Exact polynomial-time convex optimization formulations for two-layer networks. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 7695–7705, 13–18 July 2020.

H. Purwins, B. Li, T. Virtanen, J. Schlüter, S.-Y. Chang, and T. Sainath. Deep learning for audio signal processing. *IEEE Journal of Selected Topics in Signal Processing*, 13(2): 206–219, 2019.

P. Savarese, I. Evron, D. Soudry, and N. Srebro. How do infinite width bounded norm networks look in function space? *Annual Conference on Learning Theory*, pages 2667–2690, 2019.

J. Serrà, S. Pascual, and C. S. Perales. Blow: a single-scale hyperconditioned flow for non-parallel raw-audio voice conversion. *Advances in Neural Information Processing Systems*, 32, 2019.

R. P. Stanley et al. An introduction to hyperplane arrangements. *Geometric Combinatorics*, 13(389-496):24, 2004.

A. M. Stuart. Uncertainty quantification in bayesian inversion. *ICM2014. Invited Lecture*, 1279, 2014.

R. Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

R. J. Tibshirani. The lasso problem and uniqueness. *Electronic Journal of Statistics*, 7: 1456–1490, 2013.

S. Vaiter, C. Deledalle, G. Peyré, J. Fadili, and C. Dossal. The degrees of freedom of the group lasso for a general design. *CoRR*, abs/1212.6478, 2012.

Y. Wang, J. Lacotte, and M. Pilanci. The hidden convex optimization landscape of regularized two-layer relu networks: an exact characterization of optimal solutions. In *International Conference on Learning Representations*, 2021.

Y. Wang, P. Chen, and W. Li. Projected Wasserstein gradient descent for high-dimensional Bayesian inference. *SIAM/ASA Journal on Uncertainty Quantification*, 10(4):1513–1532, 2022a.

Y. Wang, P. Chen, M. Pilanci, and W. Li. Optimal neural network approximation of Wasserstein gradient direction via convex optimization. *arXiv:2205.13098*, 2022b.

O. Zahm, T. Cui, K. Law, A. Spantini, and Y. Marzouk. Certified dimension reduction in nonlinear Bayesian inverse problems. *Mathematics of Computation*, 91(336):1789–1835, 2022.

Appendix

Note: Figures 7, 15, and 16, we have $\mathbf{y} = \mathbf{h}^{(T)}$, $N = 40, T = 10$, and vectors $\mathbf{v} = (v_1, \cdots, v_N)$ are depicted by plotting (n, v_n) as a dot. Slanted lines in Figures 1, 5, 13, and 14 have slope ± 1.

Appendix A. Definitions and preliminaries

A.1 Activation function $\sigma(x)$

We assume the activation σ is *piecewise linear around* 0, i.e., of the form $\sigma(x) = \begin{cases} c_1 x + d_1 & \text{if } x < 0 \\ c_2 x + d_2 & \text{if } x \geq 0 \end{cases}$, for some $c_1, c_2, d_1, d_2 \in \mathbb{R}$. Leaky ReLu, absolute value, ReLu, sign, and threshold activations (Section 1.2) are piecewise linear around 0.

A function f is *bounded* if there is $M \geq 0$ with $|f(x)| \leq M$ for all x. If $\sigma(x)$ is piecewise linear around zero, $\sigma(x)$ is *bounded* if and only if $c_1 = c_2 = 0$, e.g. $\sigma(x)$ is a threshold or sign activation. We call f *symmetric* if it is an even or odd function, for example absolute value. The activation $\sigma(x)$ is defined to be *homogeneous* if for any $a \in \mathbb{R}^+$, $\sigma(ax) = a\sigma(x)$. Homogeneous activations include ReLU, leaky ReLU, and absolute value. They lack an amplitude parameter s. We say $\sigma(x)$ is *sign-determined* if its value depends only on the sign of its input and not its magnitude. Threshold and sign activations are sign-determined.

A.2 Effective depth

Remark 5 *Let the* inner parameters *be* $\mathbf{s}^{(i,l)}$ *where* $l \leq L-2$ *and* $\mathbf{W}^{(i,l)}$ *where* $l \leq L-1$ *for a parallel network; and* $\left(\mathbf{s}^{\mathbf{u}\oplus 1}, \cdots \mathbf{s}^{\mathbf{u}\oplus m_{L-l}}\right), \left(\alpha^{\mathbf{u}\oplus 1}, \cdots \alpha^{\mathbf{u}\oplus m_{L-l}}\right)$ *where* $\mathbf{u}$ *has positive length, and* $\left(\mathbf{w}^{\mathbf{u}\oplus 1}, \cdots \mathbf{w}^{\mathbf{u}\oplus m_{L-l}}\right)$ *for a tree network. By plugging* (4) *and* (5) *into themselves for parallel and tree networks, respectively,*

$$\mathbf{X}^{(i,l+2)} = \sigma_{\mathbf{s}^{(i,l+1)}}\left(\sigma\left(\mathbf{X}^{(i,l)}\mathbf{W}^{(i,l+1)} + \mathbf{b}^{(i,l)}\right)\mathbf{s}^{(i,l)}\mathbf{W}^{(i,l+1)} + \mathbf{b}^{(i,l+1)}\right)$$

where $1 \leq l \leq L-2$ *for a parallel network and* $\mathbf{X}^{(\mathbf{u})}=$

$$\sum_{i=1}^{m_{L-l}} \alpha^{(\mathbf{u}\oplus i)}\sigma_{s^{(\mathbf{u}\oplus i)}}\left(b^{(\mathbf{u}\oplus i)} + \sum_{j=1}^{m_{L-l-1}} \alpha^{(\mathbf{u}\oplus i\oplus j)}\sigma\left(\mathbf{X}^{(\mathbf{u}\oplus i\oplus j)}\mathbf{w}^{(\mathbf{u}\oplus i\oplus j)} + b^{(\mathbf{u}\oplus i\oplus j)}\right)s^{(\mathbf{u}\oplus i\oplus j)}\mathbf{w}^{(\mathbf{u}\oplus i)}\right),$$

where $0 \leq l \leq L-3$ *for a tree network. Suppose* σ *is sign-determined. Since the inner parameters do not affect* $f_L(\mathbf{X};\theta)$*, regularizing them will drive them to zero. We define the minimum value in* (1) *as an infimum which is approached as their* $l_{\tilde{L}}$ *norms approach* 0*. Under this definition, we can remove the inner parameters from regularization, and optimize for their values normalized by their* $l_{\tilde{L}}$*-norms, equivalently fixing their* $l_{\tilde{L}}$*-norms.*

Remark 6 *The magnitudes of the inner parameters affect the neural net output for ReLU, leaky ReLU and absolute value activations. However by Remark 5, this does not hold for sign and threshold activations. This motivates the definition for* $\tilde{L}$*.*

Appendix B. Parallel and tree networks with data dimension $d \geq 1$

Since $\xi \notin \theta_w$, the training problem (1) can be written as $\min_{\theta - \{\xi\}} \frac{\beta}{2}\|\theta_w\|_2^2 + \min_{\xi} \{\mathcal{L}_{\mathbf{y}}(f_L(X) - \xi\mathbf{1} + \xi\mathbf{1})\}$. Apply the change of variables $\theta \to \theta - \{\xi\}$, $f_L(\mathbf{X};\theta) \to f_L(\mathbf{X};\theta) - \xi\mathbf{1}, \mathcal{L}_{\mathbf{y}}(\mathbf{z}) \to \min_{\xi} L_y(\mathbf{z} + \xi\mathbf{1})$ in (1). In other words, $\mathcal{L}_{\mathbf{y}}$ absorbs ξ, which is now ommitted from $f_L(\mathbf{X};\theta)$. Note that convexity is still preserved.

Definition 8 A *rescaled* neural network and its parameter set is

$$\begin{aligned} f_L(\mathbf{x};\theta) &= \xi + \sum_{i=1}^{m_L} \alpha_i \left(\mathbf{X}^{(i,L)} \prod_{l=1}^{L-1} q^{(i,l)} \right) \\ \mathbf{X}^{(i,l+1)} &= \sigma\left(\mathbf{X}^{(i,l)} \mathbf{W}^{(i,l)} + \mathbf{b}^{(i,l)} \right), \text{ for } l \in [L-1] \\ \theta_w^{(i)} &= \left\{ \alpha_i, q^{(i,l)} : l \in [L-1] \right\}, \theta_b^{(i)} = \left\{ \mathbf{b}^{(i,l)}, \mathbf{W}^{(i,l)} : l \in [L-1] \right\} \text{ for } i \in [m_L] \end{aligned} \tag{16}$$

where $q^{(i,l)} \in \mathbb{R}$, $\left\|\mathbf{W}^{(i,l)}\right\|_L = 1$ for a parallel network with homogeneous activation,

$$\begin{aligned} f_L(\mathbf{x};\theta) &= \xi + \sum_{i=1}^{m_L} \alpha_i s^{(i,L-1)} \mathbf{X}^{(i,L)} \\ \mathbf{X}^{(i,l+1)} &= \sigma\left(\mathbf{X}^{(i,l)} \mathbf{W}^{(i,l)} + \mathbf{b}^{(i,l)} \right), l \in [L-1] \\ \theta_w^{(i)} &= \left\{ \alpha_i, s^{(i,L-1)} \right\}, \theta_b^{(i)} = \left\{ \mathbf{b}^{(i,l)}, \mathbf{W}^{(i,l)} : l \in [L-1] \right\} \text{ for } i \in [m_L] \end{aligned} \tag{17}$$

for a parallel network with sign-determined activation, and

$$\begin{aligned} f_L(\mathbf{x};\theta) &= \xi + \sum_{i=1}^{m_L} \alpha_i s^{(i)} \sigma\left(\mathbf{X}^{(i)} + b^{(i)} \mathbf{1} \right) \\ \mathbf{X}^{(\mathbf{u})} &= \begin{cases} \sum_{i=1}^{m_{L-l}} \alpha^{(\mathbf{u}\oplus i)} \sigma\left(\mathbf{X}^{(\mathbf{u}\oplus i)} + b^{(\mathbf{u}\oplus i)} \right) & \text{if } 1 \leq l \leq L-3 \\ \sum_{i=1}^{m_{L-l}} \alpha^{(\mathbf{u}\oplus i)} \sigma\left(\mathbf{X}\mathbf{w}^{(\mathbf{u}\oplus i)} + b^{(\mathbf{u}\oplus i)} \right) & \text{if } l = L-2 \end{cases} \\ \theta_w^{(i)} &= \left\{ \alpha^{(i)}, \mathbf{s}^{(i)} \right\} \theta_b^{(i)} = \left\{ \alpha^{(\mathbf{u})}, b^{(\mathbf{u})} : \mathbf{u} \in \mathcal{U}, u_1 = i \right\} \text{ for } i \in [m_L] \end{aligned} \tag{18}$$

where the length of $\mathbf{u}$ is > 1 for $\alpha^{(\mathbf{u})} \in \theta_b^{(i)}$, for a tree network with sign-determined activation.

Lemma 7 *The training problem remains equivalent if the neural network is rescaled.*

Proof *For parallel networks with homogenous activation*: Let $q^{(i,l)} = \|\mathbf{W}^{(i,l)}\|_L$. For parallel networks with homogeneous activations, for $l \in [L-1]$, by a change of variables $\mathbf{b}^{(i,l)} \to q^{(i,l)}\mathbf{b}^{(i,l)}$, the training problem is equivalent if we factor out $q^{(i,l)}$ so that

$$\mathbf{X}^{(i,l+1)} = \sigma\left(\mathbf{X}^{(i,l)} \tilde{\mathbf{W}}^{(i,l)} + \mathbf{1} \cdot \mathbf{b}^{(i,l)} \right) q^{(i,l)}, \tag{19}$$

such that $\left\|\tilde{\mathbf{W}}^{(i,l)}\right\|_{\tilde{L}} = 1$. Plug in (19) into itself for $l = 1, \cdots, L-1$ to move all the $q^{(i,l)}$ terms to $\mathbf{X}^{(i,L)}$ and a change of variables for $\mathbf{b}^{(i,l)}$ to get the result.

For parallel networks with sign-determined activation: By Remark 5, $\mathbf{s}^{(i,l)}$ for $l \leq L-2$ and $\mathbf{W}^{(i,l)}$ for $l \leq L-1$ can be unregularized. Then apply a change of variables $\mathbf{s}^{(i,l-1)}\mathbf{W}^{(i,l)} \to \mathbf{W}^{(i,l)}$ for $2 \leq l \leq L-1$, which removes $\mathbf{s}^{(i,l)}$ from θ for $l \leq L-2$.

For a tree network with sign-determined activation: Remove tree parameters from regularization as described in in Remark 5. For $\mathbf{u}$ of length $1 \leq l \leq L-2$, $\mathbf{w}^{(\mathbf{u})} \in \mathbb{R}$ and so for $i \in [m_{L-l-1}]$ we may apply a change of variables $\alpha^{(\mathbf{u}\oplus i)}s^{(\mathbf{u}\oplus i)}\mathbf{w}^{(\mathbf{u})} \to \alpha^{(\mathbf{u}\oplus i)}$, so $\mathbf{w}^{(\mathbf{u})}, s^{(\mathbf{u}\oplus i)}$ can be removed from θ. ■

For sign-determined activations, Lemma 7 still holds with the constraint $\left\|\mathbf{W}^{(i,l)}\right\|_L = 1$. Henceforth, tree networks are assumed to have sign-determined activation.

Remark 7 *We extend row-wise the recursive definitions* (3), (4), *and* (5) *to the cases where* $\mathbf{X}^{(1)}, \mathbf{X}^{(i,1)}$, *and* $\mathbf{X}^{(u_1,\cdots,u_{L-1})}$ *is* $\mathbf{X} \in \mathbb{R}^{N\times d}$, *respectively.*

Definition 9 Let $\mathbf{X}^{(i,1)}=\mathbf{X}^{(u_1,\cdots,u_{L-1})}=\mathbf{X}$. Let $\tilde{\mathbf{X}}^{(i)} \in \mathbb{R}^N$ be $\mathbf{X}^{(i,L)}$ for a rescaled parallel network or $\sigma\left(\mathbf{X}^{(i)} + b^{(i)}\mathbf{1}\right)$ for a rescaled tree network. The *rescaled* training problem is

$$\min_{\theta\in\Theta} \mathcal{L}_{\mathbf{y}}\left(\sum_{i=1}^{m_L} \alpha_i \tilde{\mathbf{X}}^{(i)}\right) + \beta\sum_{i=1}^{m_L}|\alpha_i|. \tag{20}$$

Lemma 8 *The training problem is equivalent to the rescaled problem.*

Proof For sign-determined activations, rename the final-layer amplitude parameters $s^{(i,L-1)}$ in parallel networks and $s^{(i)}$ for tree networks as $q^{(i,L-1)}$. By the above lemmas, in all cases, the training problem is equivalent to

$$\min_{\theta\in\Theta} \mathcal{L}_{\mathbf{y}}\left(\sum_{i=1}^{m_L}\left(\tilde{\mathbf{X}}^{(i)}\alpha_i \prod_{l=1}^{\tilde{L}-1} q^{(i,l)}\right)\right) + \frac{\beta}{\tilde{L}}\|\theta_w\|_{\tilde{L}}^{\tilde{L}}. \tag{21}$$

Observe that the number of regularized parameters is $|\theta_w| = \tilde{L}$. By the AM-GM inequality,

$$\frac{1}{\tilde{L}}\|\theta_w\|_{\tilde{L}}^{\tilde{L}} = \sum_{i=1}^{m_L}\frac{|\alpha_i|^{\tilde{L}} + \sum_{l=1}^{\tilde{L}-1}\left(q^{(i,l)}\right)^{\tilde{L}}}{\tilde{L}} \geq \sum_{i=1}^{m_L}\left(|\alpha_j|^{\tilde{L}}\prod_{l=1}^{\tilde{L}-1}\left(q^{(i,l)}\right)^{\tilde{L}}\right)^{1/\tilde{L}} = \sum_{i=1}^{m_L}\prod_{q\in\theta_w^{(i)}}|q|. \tag{22}$$

So a lower bound on (21) is

$$\min_{\theta\in\Theta} \mathcal{L}_{\mathbf{y}}\left(\sum_{i=1}^{m_L}\left(\tilde{\mathbf{X}}^{(i)}\mathrm{sign}(\alpha_i)\prod_{q\in\theta_w^{(i)}}|q|\right)\right) + \beta\sum_{i=1}^{m_L}\prod_{q\in\theta_w^{(i)}}|q|. \tag{23}$$

Letting $\gamma^{(i)} = \left(\prod_{q\in\theta_w^{(i)}}|q|\right)^{1/\tilde{L}}$ for $i \in [m_{L-1}]$ and making $q \to \mathrm{sign}\,(q)\,\gamma^{(i)}$ for $q \in \theta_w^{(i)}$ in (23) makes the objective of (23) the same as in (21). So, (21) and (23) are equivalent. Finally, for each $j \in [m_{L-1}]$, apply a change of variables $\prod_{q\in\theta_w^{(i)}}|q| \to \alpha_i$ so that (23) becomes (20). Also, $q^{(i,l)}$ can be removed from θ. ■

Lemma 9 *A lower bound on the rescaled training problem* (20)*is*

$$\max_{\lambda\in\mathbb{R}^N} -\mathcal{L}^*_{\mathbf{y}}(\lambda) \quad s.t. \quad \max_{\theta\in\Theta}\left|\lambda^T\tilde{\mathbf{X}}\right| \leq \beta, \tag{24}$$

where $\tilde{\mathbf{X}} = \tilde{\mathbf{X}}^{(1)}$ *and* $f^*(\mathbf{x}) := \max_{\mathbf{x}}\left\{\mathbf{z}^T\mathbf{x} - f(\mathbf{x})\right\}$ *is the convex conjugate of* f*.*

Proof Find the dual of (20), by rewriting (20) as

$$\min_{\theta\in\Theta}\mathcal{L}_{\mathbf{y}}(\mathbf{z}) + \beta||\boldsymbol{\alpha}||_1, \quad \text{s.t.} \quad \mathbf{z} = \sum_{i=1}^{m_L}\alpha_i\tilde{\mathbf{X}}^{(i)}. \tag{25}$$

The Lagrangian of problem (25) is $L(\lambda,\theta) = \mathcal{L}_{\mathbf{y}}(\mathbf{z}) + \beta||\boldsymbol{\alpha}||_1 - \lambda^T\mathbf{z} + \sum_{i=1}^{m_L}\lambda^T\tilde{\mathbf{X}}^{(i)}\alpha_i$. Minimize the Lagrangian over $\mathbf{z}$ and $\boldsymbol{\alpha}$ and use Fenchel duality Boyd and Vandenberghe (2004). The dual of (25) is

$$\max_{\lambda\in\mathbb{R}^N} -\mathcal{L}^*_{\mathbf{y}}(\lambda) \quad \text{s.t.} \quad \max_{\theta\in\Theta}\left|\lambda^T\tilde{\mathbf{X}}^{(i)}\right| \leq \beta, i\in[m_L]. \tag{26}$$

In the tree and parallel nets, $\tilde{\mathbf{X}}^{(i)}$ is of the same form for all $i\in[m_L]$. So the m_L constraints in (26) collapse to just one constraint. Then we can write (26) as (24). ■

If the network has a parallel architecture, let $\mathbf{X}^{(l)}$ be defined as in (3), where $\mathbf{X}^{(1)} = \mathbf{X}$ (Remark 7). This makes the lower bound problem (24) for a parallel network equivalent to

$$\max_{\lambda\in\mathbb{R}^N} -\mathcal{L}^*_{\mathbf{y}}(\lambda) \quad \text{s.t.} \quad \max_{\theta\in\Theta}\left|\lambda^T\mathbf{X}^{(L)}\right| \leq \beta. \tag{27}$$

Appendix C. Deep narrow networks with data dimension $d = 1$

In this section, assume $d = 1$ and the neural net is a deep narrow network. Let $w^{(l)} = \mathbf{W}^{(l)} \in \{-1, 1\}$ and $b^{(l)} = \mathbf{b}^{(l)} \in \mathbb{R}$. Then $\mathbf{X}^{(l+1)} = \sigma\left(\mathbf{X}^{(l)}w^{(l)} + b^{(l)}\mathbf{1}\right) \in \mathbb{R}^N$. In the next results involving (27), let c_1, c_2 be defined as in Appendix A.1.

Remark 8 *Let* $b = b^{(L-1)}, a_n = \mathbf{X}^{(L-1)}_n w^{(L-1)}, g_n(b) = \sigma(a_n + b)$*. Let* $g(b) = \sum_{n=1}^N \lambda_n g_n(b) = \lambda^T\mathbf{X}^{(L)}$*. Let* $\mathcal{I}_n$ *be the set of breakpoints of* g_n *and* $\mathcal{I} = \bigcup_{n=1}^N \mathcal{I}_n$*, which contains the breakpoints of* g*. Observe* $g(b) = \sum_{n=1}^N \lambda_n c(a_n + b) = cb\sum_{n=1}^N \lambda_n + \sum_{n=1}^N \lambda_n a_n c$ *for* b *large enough (with* $c = c_2$*) and for* b *small enough (with* $c = c_1$*). So* $c_1 = c_2 = 0$ *or* $\sum_{n=1}^N \lambda_n = 0$ *if and only if* g *is bounded, if and only if* g *has a (finite) maximizer and minimizer. In this case, assuming* g *is not a constant function,* $\mathcal{I}$ *contains a maximizer and minimizer of* g*.*

Let $R_{(a,b)}$ denote reflections as defined in Definition 2.

Lemma 10 *Suppose* $\lambda^T\mathbf{1} = 0$ *if* $c_1 \neq 0$ *or* $c_2 \neq 0$*. Consider a deep narrow network with* $L > 2$ *only for ReLU activation, and* $d = 1$*. For* $l \in \{L-1, L-2, L-3\}$*, there is* $n^{(l)} \in [N]$ *such that* $b^{(l)} = -\mathbf{X}^{(l)}_{n^{(l)}}w^{(l)}$ *or* $b^{(L-3)} = w^{(L-3)}R_{\left(\mathbf{X}^{(L-3)}_{n^{(L-3)}}, \mathbf{X}^{(L-3)}_{n^{(L-2)}}\right)}$ *are optimal* $b^{(l)}$ *values in the maximization constraint in* (27)*.*

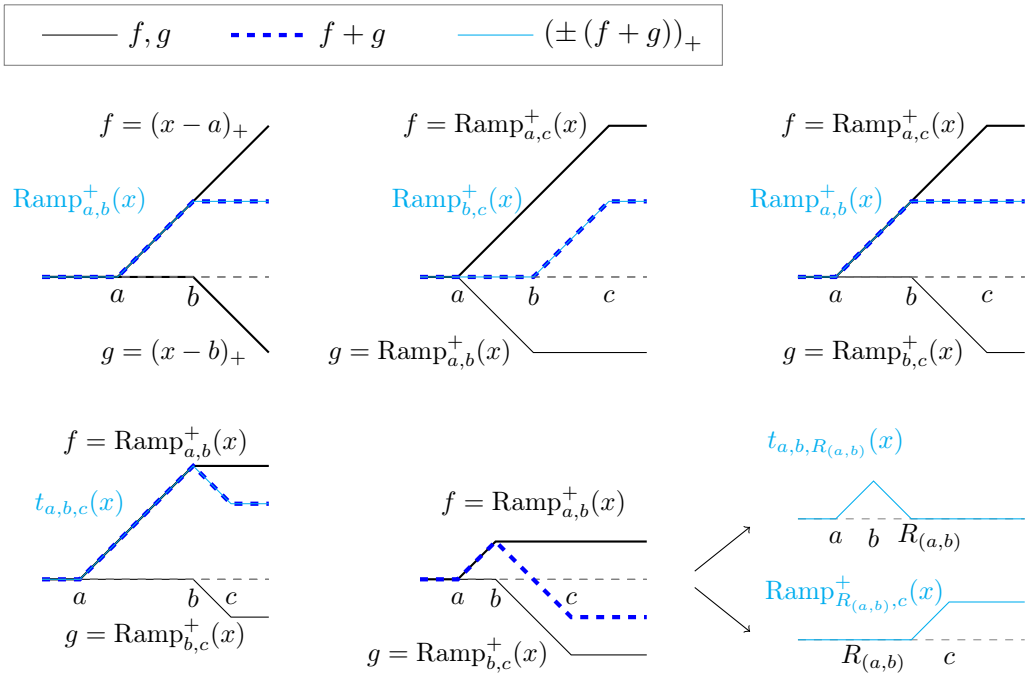


Figure 13: Sums, scalings, and ReLU's applied to capped ramps. For $a, b, c \in \mathbb{R}$, the function $t_{a,b,c}(x)$ is 0 if $x \leq a$, $x - a$ if $x \in [a, b]$, $R_{(a,b)} - x$ if $x \in [b, c]$ and $R_{(a,b)} - c$ if $x \geq c$.

Proof The dual constraint's objective is $\lambda^T \mathbf{X}^{(L)} = \sum_{n=1}^{N} \lambda_n \sigma\left(\mathbf{X}_n^{(L-1)} w^{(L-1)} + b^{(L-1)}\right)$. The breakpoints of $g_n\left(b^{(L-1)}\right) = \sigma\left(\mathbf{X}_n^{(L-1)} w^{(L-1)} + b^{(L-1)}\right)$ occur where they make the argument of an activation zero, and by Remark 8 and the assumption involving $\lambda^T \mathbf{1} = 0$, these breakpoints contain an optimal $b^{(L-1)}$. Therefore for some $n^{(L-1)} \in [N]$, $-\mathbf{X}_{n^{(L-1)}}^{(L-1)} w^{(L-1)}$ is an optimal $b^{(L-1)}$. Plugging in this optimal $b^{(L-1)}$ makes $\lambda^T \mathbf{X}^{(L-1)} =$

$$\begin{aligned} &\sum_{n=1}^{N} \lambda_n \sigma\left(w^{(L-1)}\left(\mathbf{X}_n^{(L-1)} - \mathbf{X}_{n^{(L-1)}}^{(L-1)}\right)\right) = \\ &\sum_{n=1}^{N} \lambda_n \sigma\left(w^{(L-1)}\left(\sigma\left(\mathbf{X}_n^{(L-2)} w^{(L-2)} + b^{(L-2)}\right) - \sigma\left(\mathbf{X}_{n^{(L-1)}}^{(L-2)} w^{(L-2)} + b^{(L-2)}\right)\right)\right). \end{aligned} \tag{28}$$

.

Now assume $\sigma(x) = \text{ReLU}(x)$. Setting $x = b^{(L-2)}, a = -\mathbf{X}_n^{(L-2)} w^{(L-2)}$, $b = -\mathbf{X}_{n^{(L-1)}}^{(L-2)} w^{(L-2)}, f = \sigma\left(\mathbf{X}_n^{(L-2)} w^{(L-2)} + b^{(L-2)}\right)$ and $g = \sigma\left(\mathbf{X}_{n^{(L-1)}}^{(L-2)} w^{(L-2)} + b^{(L-2)}\right)$, the top right plot of Figure 13 shows that as a function of $b^{(L-2)}$, for all $n \in [N]$, $\mathbf{X}_n^{(L)}$ is bounded and has breakpoints of the form $-\mathbf{X}_{n^{(L-2)}}^{(L-2)} w^{(l-2)}$ for $n^{(L-2)} \in [N]$. So by Remark 8, there exists $n^{(L-2)} \in [N]$ for which $b^{(L-2)} = -\mathbf{X}_{n^{(L-2)}}^{(L-2)} w^{(l-2)}$ is optimal. Plugging in this

optimal $b^{(L-2)}$ into (28) makes $\lambda^T \mathbf{X}^{(L)} =$

$$\sum_{n=1}^{N} \lambda_n \sigma\left(w^{(L-1)}\left(\sigma\left(w^{(L-2)}\left(\mathbf{X}_n^{(L-2)} - \mathbf{X}_{n^{(L-2)}}^{(L-2)}\right)\right) - \sigma\left(w^{(L-2)}\left(\mathbf{X}_{n^{(L-1)}}^{(L-2)} - \mathbf{X}_{n^{(L-2)}}^{(L-2)}\right)\right)\right)\right) =$$
$$\sum_{n=1}^{N} \lambda_n \sigma\left(w^{(L-1)}\left(\sigma\left(w^{(L-2)}\left(\sigma\left(w^{(L-3)}\mathbf{X}_n^{(L-3)} + b^{(L-3)}\right) - \sigma\left(w^{(L-3)}\mathbf{X}_{n^{(L-2)}}^{(L-3)} + b^{(L-3)}\right)\right)\right)\right.\right.$$
$$\left.\left. - \sigma\left(w^{(L-2)}\left(\sigma\left(w^{(L-3)}\mathbf{X}_{n^{(L-1)}}^{(L-3)} + b^{(L-3)}\right) - \sigma\left(w^{(L-3)}\mathbf{X}_{n^{(L-2)}}^{(L-3)} + b^{(L-3)}\right)\right)\right)\right)\right).$$

Setting $x = b^{(L-3)}$, the first plot of Figure 13 graphs the difference of ReLU functions and shows that $\sigma\left(w^{(L-2)}\left(\sigma\left(w^{(L-3)}\mathbf{X}_n^{(L-3)} + b^{(L-3)}\right) - \sigma\left(w^{(L-3)}\mathbf{X}_{n^{(L-2)}}^{(L-3)} + b^{(L-3)}\right)\right)\right)$ and $\sigma\left(w^{(L-2)}\left(\sigma\left(w^{(L-3)}\mathbf{X}_{n^{(L-1)}}^{(L-3)} + b^{(L-3)}\right) - \sigma\left(w^{(L-3)}\mathbf{X}_{n^{(L-2)}}^{(L-3)} + b^{(L-3)}\right)\right)\right)$ are ramps. The rest of the plots in Figure 13 graph the difference of ramps the cyan plots show that as a function of $b^{(L-3)}$, $\mathbf{X}_n^{(L)}$ is bounded and has breakpoints of the form $b^{(L-3)} = -\mathbf{X}_{n^{(L-3)}}^{(L-3)} w^{(l-3)}$ or $b^{(L-3)} = w^{(L-3)} R_{\left(\mathbf{X}_{n^{(L-3)}}^{(L-3)}, \mathbf{X}_{n^{(L-2)}}^{(L-3)}\right)}$ for some $n^{(L-3)} \in [N]$. By Remark 8, there exist points of this form for the optimal $b^{(L-3)}$. ■

Let $\mathcal{M}^{(2)}, \mathcal{M}^{(3)}, \mathcal{M}$ be dictionary index sets as defined in Definition 3.

Definition 10 Let $\tilde{\mathcal{M}}^{(1)} = \{-1, 1\}^{L-1}, \quad \tilde{\mathcal{M}} = \tilde{\mathcal{M}}^{(1)} \times \mathcal{M}^{(2)} \times \mathcal{M}^{(3)}$.

Given $(\mathbf{s}, \mathbf{j}, k) \in \tilde{\mathcal{M}}$, recursively define the *recursive dictionary function* $\tilde{f}_{\mathbf{s},\mathbf{j},k}^{(l+1,\sigma)}(x) = \sigma\left(s_l\left(\tilde{f}_{\mathbf{s},\mathbf{j},k}^{(l,\sigma)}(x) - \tilde{f}_{\mathbf{s},\mathbf{j},k}^{(l,\sigma)}\left(\tilde{a}^{(l)}\right)\right)\right)$ where $\tilde{f}_{\mathbf{s},\mathbf{j},k}^{(1,\sigma)}(x) = x$ and $\tilde{a}^{(l)} = \begin{cases} R_{(x_{j_1}, x_{j_2})} \text{ if } l = 1, k = 1 \\ x_{j_l} \text{ else.} \end{cases}$.

Remark 9 *For $l \in [3]$, the $(3l)^{th}$ rows of Figure 14 list and plot possible graphs of $\tilde{f}_{\mathbf{s},\mathbf{j},k}^{(l+1,\sigma)}$.*

Lemma 11 *For $L = 2, 3, 4$ layers, where $L \in \{3, 4\}$ only if the activation is ReLU, the maximization constraint in (27) is equivalent to*

$$\forall (\mathbf{s}, \mathbf{j}, k) \in \tilde{\mathcal{M}}, \quad \left|\sum_{n=1}^{N} \lambda_n \tilde{f}_{\mathbf{s},\mathbf{j},k}^{(L,\sigma)}(x_n)\right| \leq \beta \qquad (29)$$
$$\mathbf{1}^T \lambda = 0 \text{ if } c_1 \neq 0 \text{ or } c_2 \neq 0.$$

Moreover, $\tilde{\mathcal{M}}$ can be replaced by $\mathcal{M}$ in (29).

Proof By Lemma 10, (27) remains equivalent if $\mathbf{X}^{(L)}$ is defined by $\mathbf{X}^{(1)} = \mathbf{X}$ and

$$\mathbf{X}_n^{(l+1)} = \begin{cases} \left(s_1\left(x_n - R_{(x_{j_1}, x_{j_2})}\right)\right)_+ & \text{if } k = 1, l = 1 \\ \left(s_l\left(\mathbf{X}_n^{(l)} - \mathbf{X}_{j_l}^{(l)}\right)\right)_+ & \text{else} \end{cases} = \tilde{f}_{\mathbf{s},\mathbf{j},k}^{(l+1,\sigma)}(x_n), \qquad (30)$$

This gives (29).

Now, if the activation is symmetric, then $\tilde{f}^{(L,\sigma)}_{\mathbf{s},\mathbf{j},k}$ is invariant under the sign of the components of $\mathbf{s}$. Next, recall $x_1 > \cdots > x_N$ and $\text{sign}(0) = 1$. If the activation is sign, then for all $n \in [N-1]$, $\tilde{f}^{(L=2)}_{\mathbf{s}=1,\mathbf{j}=n,k=0}(\mathbf{X}) = \sigma(\mathbf{X} - x_n) = \left(\mathbf{1}^T_{1:n}, -\mathbf{1}^T_{n+1:N}\right) = -\sigma(x_{n+1} - \mathbf{X}) = -\tilde{f}^{(L=2)}_{\mathbf{s}=-1,\mathbf{j}=n+1,k=0}$. And $\tilde{f}^{(L=2)}_{\mathbf{s}=1,\mathbf{j}=N,k=0}(\mathbf{X}) = \mathbf{1} = \tilde{f}^{(L=2)}_{\mathbf{s}=-1,\mathbf{j}=1,k=0}(\mathbf{X})$. So for $L=2$ with symmetric or sign activation, (29) is unchanged if $\mathbf{s} \in \{-1, 1\}$ is restricted to be 1, and therefore $\tilde{\mathcal{M}}$ can be replaced by $\mathcal{M}$. ■

Lemma 12 *Let* $\mathbf{A}$ *be a matrix with columns* $\tilde{f}^{(L,\sigma)}_{\mathbf{s},\mathbf{j},k}(\mathbf{X})$ *for all* $(\mathbf{s},\mathbf{j},k) \in \mathcal{M}$ *. Replace the maximization constraint in* (27) *with* (29)*. The dual of* (27) *then is*

$$\min_{\mathbf{z},\xi\in\mathbb{R}} \mathcal{L}_{\mathbf{y}}(\mathbf{A}\mathbf{z} + \xi\mathbf{1}) + \beta\|\mathbf{z}\|_1, \quad \textit{where} \quad \xi = 0 \textit{ if } c_1 = c_2 = 0. \tag{31}$$

Proof Problem (27) can be written as

$$-\min_{\lambda\in\mathbb{R}^N} \mathcal{L}^*_{\mathbf{y}}(\lambda) \quad \text{s.t.} \quad \lambda^T\mathbf{1} = 0 \text{ if } c_1 \neq 0 \text{ or } c_2 \neq 0, \text{ and } |\lambda^T\mathbf{A}| \leq \beta\mathbf{1}^T. \tag{32}$$

The Lagrangian of the negation of (32) with bidual variables $\mathbf{z}, \xi$ is

$$L(\lambda, \mathbf{z}, \xi) = \mathcal{L}^*_{\mathbf{y}}(\lambda) - \lambda^T(\mathbf{A}\mathbf{z} + \xi\mathbf{1}) - \beta\|\mathbf{z}\|_1, \text{ where } \xi = 0 \text{ if } c_1 = c_2 = 0. \tag{33}$$

Equation (33) holds because the constraint $|\lambda^T\mathbf{A}| \leq \beta\mathbf{1}^T$ i.e., $\lambda^T\mathbf{A} - \beta\mathbf{1}^T \leq \mathbf{0}^T, -\lambda^T\mathbf{A} - \beta\mathbf{1}^T \leq \mathbf{0}^T$, appears in the Lagrangian as $\lambda^T\mathbf{A}\left(\mathbf{z^{(1)}} - \mathbf{z^{(2)}}\right) - \beta\mathbf{1}^T\left(\mathbf{z^{(1)}} + \mathbf{z^{(2)}}\right)$ with bidual variables $\mathbf{z^{(1)}}, \mathbf{z^{(2)}}$, which are combined into one bidual variable $\mathbf{z} = \mathbf{z^{(1)}} - \mathbf{z^{(2)}}$. This makes $\mathbf{z^{(1)}} + \mathbf{z^{(2)}} = \|\mathbf{z}\|_1$. Changing variables $\mathbf{z}, \xi \to -\mathbf{z}, -\xi$ gives (33). Since $\mathcal{L}^{**} = \mathcal{L}$ Borwein and Lewis (2000), $\inf_\lambda L(\lambda, \mathbf{z}, \xi) = -\mathcal{L}_y - \beta\|\mathbf{z}\|_1$ and negating its maximization gives (31). ■

Remark 10 *Let* $a_1 \in \{x_1, \cdots, x_N\} \cup \bigcup_{j_1,j_2\in[N]} \left\{R_{(x_{j_1}, x_{j_2})}\right\}$ *and let* $a_2 = a^{(i)}_2, a_3 = a^{(i)}_3$ *as defined in Definition 6. By case analysis on the arguments of ReLU, the legend labeling the breakpoints in Figure 14 simplifies to the legend in Figure 5. In Definition 10, if* $l = 1$ *then* $\tilde{a}^{(1)} = a_1$ *as defined in Definition 4. By Definition 6,* $a_3 \leq a_2$*, so the second and fifth branches in the sixth row of arrows of Figure 14 do not occur. Therefore the* $(3l)^{th}$ *rows of Figure 14 for* $l \in [3]$ *constitute Figure 5. Remark 9 and simplifying the legend in Figure 5 gives* $\tilde{f}^{(l,\sigma)}_{\mathbf{s},\mathbf{j},k} = f^{(l,\sigma)}_{\mathbf{s},\mathbf{j},k}$ *as defined in Definition 4.*

Remark 11 *In Remark 10, the observations made in Figure 5 to get* $\tilde{f}^{(l,\sigma)}_{\mathbf{s},\mathbf{j},k} = f^{(l,\sigma)}_{\mathbf{s},\mathbf{j},k}$ *are the following.*

If $s_2 = 1, s_3 = -1$*:*

$$f^{(l,\sigma)}_{\mathbf{s},\mathbf{j},k} = \begin{cases} Ramp^{+}_{\min\{a,x_{j_2},x_{j_3}\},\min\{a,x_{j_2}\}} & \textit{if } s_1 = -1 \\ Ramp^{-}_{\max\{a,x_{j_2}\},\max\{a,x_{j_2},x_{j_3}\}} & \textit{if } s_1 = 1 \end{cases}$$

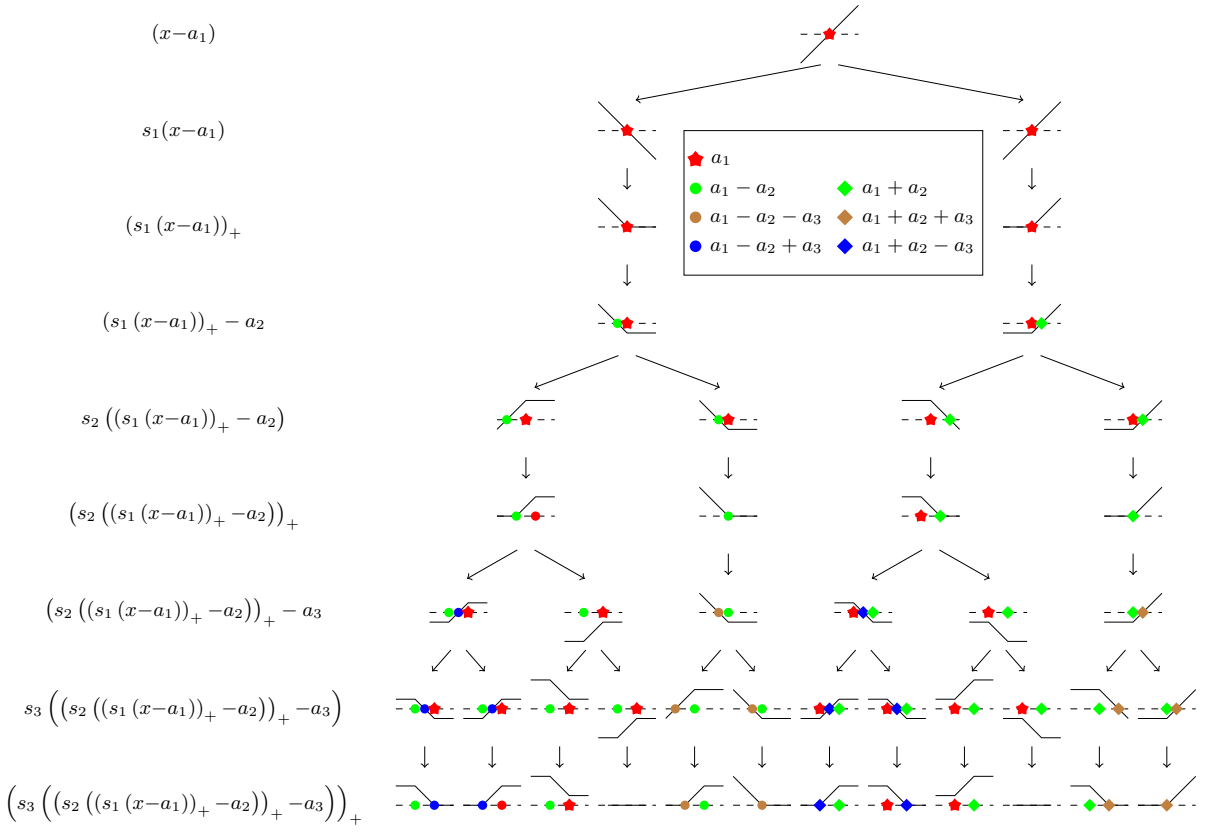


Figure 14: For $l \in \{0, 1, 2\}$, the $(3l + 1)^{\text{th}}$ row of arrows point to the right and left for the cases $s_{l+1} \geq 0$ or $s_{l+1} \leq 0$, respectively. The sixth row of arrows point to the right or left for the cases $c \geq b$ or $c \leq b$, respectively.

If $s^{(2)} = -1, s^{(3)} = 1$*:*

$$f^{(l,\sigma)}_{\mathbf{s},\mathbf{j},k} = \begin{cases} Ramp^{+}_{\min\{a,\max\{x_{j_3},\min\{a,x_{j_2}\}\}\},a} & \text{if } s_1 = -1 \\ Ramp^{-}_{a,\max\{a,\min\{x_{j_3},\max\{a,x_{j_2}\}\}\}} & \text{if } s_1 = 1. \end{cases}$$

If $s^{(2)} = -1, s^{(3)} = -1$*:*

$$f^{(l,\sigma)}_{\mathbf{s},\mathbf{j},k} = \begin{cases} Ramp^{-}_{\min\{a,x_{j_2}\},\min\{a,\max\{x_{j_3},\min\{a,x_{j_2}\}\}\}} & \text{if } s^{(1)} = -1 \\ Ramp^{+}_{\max\{a,\min\{x_{j_3},\max\{a,x_{j_2}\}\}\},\max\{a,x_{j_2}\}} & \text{if } s^{(1)} = 1. \end{cases}$$

Remark 12 *In Remark 10, the labels of the colored breakpoints in Figure 14 and Figure 5 are equivalent by the following simplifications.*

- $a_1 - a_2 = a_1 - (a_1 - x_{j_2})_+ = \begin{cases} x_{j_2}, & \text{if } a_1 > x_{j_2} \\ a_1 & \text{else} \end{cases}$
  $= \min\{a_1, x_{j_2}\}$
- $a_1 + a_2 = a_1 + (x_{j_2} - a_1)_+ = \begin{cases} x_{j_2}, & \text{if } a_1 < x_{j_2} \\ a_1 & \text{else} \end{cases}$
  $= \max\{a_1, x_{j_2}\}$
- $a_1 - a_2 - a_3 = a_1 - a_2 - \left((a_1 - x_{j_3})_+ - a_2\right)_+$
  $= \begin{cases} x_{j_3} & \text{if } a_1 - x_{j_3} > a_2 \\ a_1 - a_2 & \text{else} \end{cases}$
  $= \min\{x_{j_3}, a_1 - a_2\} = \min\{a_1, x_{j_2}, x_{j_3}\}$
- $a_1 + a_2 + a_3 = a_1 + a_2 + \left((x_{j_2} - a_1)_+ - a_2\right)_+$
  $= \begin{cases} x_{j_3} & \text{if } x_{j_3} - a_1 > a_2 \\ a_1 + a_2 & \text{else} \end{cases}$
  $= \max\{x_{j_3}, a_1 + a_2\} = \max\{a_1, x_{j_2}, x_{j_3}\}$
- $a_1 - a_2 + a_3 = a_1 - a_2 - \left(-(a_1 - x_{j_3})_+ + a_2\right)_+$
  $= \begin{cases} x_{j_3} & \text{if } 0 < a_1 - x_{j_3} < a_2 \\ a_1 - a_2 & \text{if } a_1 - x_{j_3} > a_2 \\ a_1 & \text{else} \end{cases}$
  $= \min\{a_1, \max\{x_{j_3}, a_1 - a_2\}\}$
  $= \min\{a_1, \max\{x_{j_3}, \min\{a_1, x_{j_2}\}\}\}$
- $a_1 + a_2 - a_3 = a_1 + a_2 - \left(-(x_{j_3} - a_1)_+ + a_2\right)_+$
  $= \begin{cases} x_{j_3} & \text{if } 0 < x_{j_3} - a_1 < a_2 \\ a_1 + a_2 & \text{if } x_{j_3} - a_1 > a_2 \\ a_1 & \text{else .} \end{cases}$
  $= \max\{a_1, \min\{x_{j_3}, a_1 + a_2\}\}$
  $= \max\{a_1, \min\{x_{j_3}, \max\{a_1, x_{j_2}\}\}\}$

Proof [Lemma 1] We simplify Definition 4 when $L = 4$ by considering the values of $\mathbf{s}$ for which $f^{(L,\sigma)}_{\mathbf{s},\mathbf{j},k}$ has a breakpoint at a reflection $a_1 = R_{(x_{j_1}, x_{j_2})}$. When $s^{(2)} = s^{(3)} = 1$, the result follows. Otherwise, $f^{(L,\sigma)}_{\mathbf{s},\mathbf{j},k}$ is

- $\text{Ramp}^{+}_{x_{j_3}, R_{(x_{j_1}, x_{j_2})}}$ when $\mathbf{s} = (-1, 1, -1)$ for any $x_{j_3} \leq R_{(x_{j_1}, x_{j_2})} \leq x_{j_2} \leq x_{j_1}$ or when $\mathbf{s} = (-1, -1, 1)$ for any $x_{j_1} \leq x_{j_2} \leq x_{j_3} \leq R_{(x_{j_1}, x_{j_2})}$.
- $\text{Ramp}^{-}_{R_{(x_{j_1}, x_{j_2})}, x_{j_3}}$ when $\mathbf{s} = (1, 1, -1)$ for any $x_{j_1} \leq x_{j_2} \leq R_{(x_{j_1}, x_{j_2})} \leq x_{j_3}$ or when $\mathbf{s} = (1, -1, 1)$ for any $x_{j_1} \geq x_{j_2} \geq x_{j_3} \geq R_{(x_{j_1}, x_{j_2})}$.
- $\text{Ramp}^{+}_{R_{(x_{j_1}, x_{j_2})}, x_{j_2}}$ when $\mathbf{s} = (1, -1, -1)$ for any $x_{j_2} \leq x_{j_1}$
- $\text{Ramp}^{-}_{R_{(x_{j_2}, x_{j_1})}, x_{j_2}}$ when $\mathbf{s} = (-1, -1, -1)$ for any $x_{j_2} \geq x_{j_1}$

The first two cases give $\text{Ramp}_{x_{j_3}, R_{(x_{j_1}, x_{j_2})}}$ and the last two cases give $\text{Ramp}_{R_{(x_{j_1}, x_{j_2})}, x_{j_2}}$. ■

Proof [Theorem 1] By Lemma 12, problem (31) is a lower bound on the training problem (1), where the Lasso features are $\tilde{f}^{(L,\sigma)}_{\mathbf{s},\mathbf{j},k}$. Let $(\mathbf{z}^*, \xi^*)$ be a Lasso solution. By the equivalent expressions for $\tilde{f}^{(L,\sigma)}_{\mathbf{s},\mathbf{j},k}$ in Figure 14 (see Remark 9), the parameters formed by the reconstruction defined in Definition 6 without unscaling achieves the same objective in the rescaled training problem, as $(\mathbf{z}^*, \xi^*)$ does in the Lasso objective. Parameter unscaling makes them achieve the same objective in the training problem (see Remark 13). By Remark 10, $\tilde{f}^{(L,\sigma)}_{\mathbf{s},\mathbf{j},k} = f^{(L,\sigma)}_{\mathbf{s},\mathbf{j},k}$ and so the Lasso problem in Theorem 1 is equivalent to the non-convex training problem. ■

Remark 13 *For sign-determined activations, by Remark 5, the inner weights can be unregularized. So, reconstructed parameters (as defined in Definition 6) that are unscaled according to either definition of unscaling in Section 2 achieve the same objective in the training problem as the optimal value of the rescaled problem.*

Appendix D. Deep neural networks with sign activation

In this section, we assume $\sigma(x) = \text{sign}(x)$. First we discuss parallel architectures.

Definition 11 Define the *hyperplane arrangement set* for a matrix $\mathbf{Z} \in \mathbb{R}^{N\times m}$ as

$$\mathcal{H}(\mathbf{Z}) := \{\sigma\left(\mathbf{Z}\mathbf{w} + b\mathbf{1}\right) : \mathbf{w} \in \mathbb{R}^m, b \in \mathbb{R}\} \subset \{-1, 1\}^N. \tag{34}$$

Let S_0 be the set of columns of $\mathbf{X}$. Let $\{S_l\}_{l=1}^{L-1}$ be a tuple of sets satisfying $S_l \subset \mathcal{H}([S_{l-1}])$ and $|S_l| = m_l$. Let $A_{Lpar}(\mathbf{X})$ be the union of all possible sets S_{L-1}.

In Definition 11, since $m_{L-1} = 1$ in a parallel network, S_{L-1} contains one vector. The set $\mathcal{H}(\mathbf{Z})$ denotes all possible $\{1, -1\}$ labelings of the samples $\{\mathbf{z}_i\}_{i=1}^N$ by a linear classifier. Its size is upper bounded by $|\mathcal{H}(\mathbf{Z})| \leq 2\sum_{k=0}^{r-1}\binom{N-1}{k} \leq 2r\left(\frac{e(n-1)}{r}\right)^r \leq 2^N$, where $r := \text{rank}(\mathbf{Z}) \leq \min(N, m)$ Cover (1965); Stanley et al. (2004).

Lemma 13 *The lower bound problem* (27) *is equivalent to*

$$\max_{\lambda} -\mathcal{L}_{\mathbf{y}}^*(\lambda), \quad \text{s.t.} \max_{\mathbf{h}\in A_{Lpar}(\mathbf{X})} |\lambda^T\mathbf{h}| \leq \beta. \tag{35}$$

Proof For $l \in [L]$, there is $S_{l-1} \subset \mathcal{H}\left(\mathbf{X}^{(l-1)}\right)$ with $\mathbf{X}^{(l)} = [S_{l-1}]$. Recursing over $l \in [L]$ gives $\left\{\mathbf{X}^{(L)} : \theta \in \Theta\right\} = A_{Lpar}(\mathbf{X})$. So, the constraints of (27) and (35) are the same. ■

Remark 14 *Without loss of generality (by scaling by -1), assume that the vectors in $\mathcal{H}(\mathbf{Z})$ start with* 1*. Under this assumption, Lemma 13 still holds.*

Lemma 14 *Let $\mathbf{A} = [A_{Lpar}(\mathbf{X})]$. The lower bound problem* (35) *is equivalent to*

$$\min_{\mathbf{z}} \quad \mathcal{L}_{\mathbf{y}}(\mathbf{A}\mathbf{z}) + \beta||\mathbf{z}||_1. \tag{36}$$

Proof Problem (35) is the dual of (36), and since the problems are convex with feasible regions that have nonempty interior, by Slater's condition, strong duality holds Boyd and Vandenberghe (2004). ■

Remark 15 *The set $A_{L,par}$ consists of all possible sign patterns at the final layer of a parallel neural net, up to multiplying by -1.*

Lemma 15 *Let $\mathbf{A}$ be defined as in Lemma 14. Let $\mathbf{z}$ be a solution to* (36)*. Suppose $m_L \geq \|\mathbf{z}\|_0$. There is a parallel neural network satisfying $f_L\left(\mathbf{X};\theta\right) = \mathbf{A}\mathbf{z}$ which achieves the same objective in the rescaled training problem* (20) *as $\mathbf{z}$ does in* (36)*.*

Proof By definition of $A_{L,par}$ (Definition 11), for every $\mathbf{A}_i \in A_L(\mathbf{X})$, there are tuples $\left\{\mathbf{W}^{(i,l)}\right\}_{l=1}^{L-1}, \left\{\mathbf{b}^{(i,l)}\right\}_{l=1}^{L-1}$ of parameters such that $\mathbf{A}_i = \mathbf{X}^{(i,L)}$. Let $\mathcal{I} = \{i : z_i \neq 0\}$. For $i \in \mathcal{I}$, set $\alpha_i = z_i$. This gives a neural net $f_L(\mathbf{X}; \theta) = \sum_{i \in \mathcal{I}} \alpha_i \mathbf{X}^{(i,L)} = \mathbf{A}\mathbf{z}$ with $|\mathcal{I}| \leq m_L$. ■

Remark 16 *Lemma 15 analogously holds for the tree network by a similar argument: by construction of A_{Ltree}, there is a neural net satisfying $f_L(\mathbf{X}; \theta) = \mathbf{A}\mathbf{z}$.*

Proposition 7 *For L-layer parallel networks with sign activation, the Lasso problem* (36) *problem and the original training problem are equivalent.*

Proof By Lemma 14, the Lasso problem is a lower bound for the training problem. By the reconstruction in Lemma 15 (see Remark 13), the lower bound is met with equality. ■

Definition 12 Recall the set $\mathcal{H}$ defined in (34). Define a matrix-to-matrix operator

$$J^{(m)}(\mathbf{Z}) := \left[\bigcup_{|S|=m} \mathcal{H}(\mathbf{Z}_S) \right]. \tag{37}$$

For $L = 2$, let $A_{L\text{tree}}(\mathbf{X}) = \mathcal{H}(\mathbf{X})$ and for $L \geq 2$, let $A_{L\text{tree}}(\mathbf{X})$ be the set of columns in $J^{(m_{L-1})} \circ \cdots \circ J^{(m_2)}(\mathcal{H}(\mathbf{X}))$.

The columns of $J^{(m)}(\mathbf{Z})$ are all hyperplane arrangement patterns of m columns of $\mathbf{Z}$.

Lemma 16 *For $L \geq 3$, the lower bound problem* (24) *for tree networks is equivalent to*

$$\max_{z \in \mathbb{R}^N} -\mathcal{L}^*_{\mathbf{y}}(\lambda), \quad \text{s.t.} \quad \max_{\mathbf{h} \in A_{L\,tree}(\mathbf{X})} |\lambda^T \mathbf{h}| \leq \beta. \tag{38}$$

Proof Let $\mathbf{u}$ be a tuple such that $u_1 = 1$. First suppose $\mathbf{u}$ has length $L-2$. For all nodes i, $\left\{\sigma\left(\mathbf{X}\mathbf{w}^{(\mathbf{u}+i)} + b^{(\mathbf{u}+i)}\mathbf{1}\right) : \mathbf{w}^{(\mathbf{u}+i)} \in \mathbb{R}^d, b^{(\mathbf{u}+i)} \in \mathbb{R}\right\} = \mathcal{H}(\mathbf{X})$ independently of any other sibling nodes $j \neq i$. So every $\mathbf{X}^{(\mathbf{u})}$ is the linear combination of m_2 columns in $\mathcal{H}(\mathbf{X})$, with the choice of columns independent of other $\mathbf{u}$ of the same length. Next, for all $\mathbf{u}$ of length $L-3$, the set of all possible $\sigma\left(\mathbf{X}^{(\mathbf{u}+i)} + b^{(\mathbf{u}+i)}\mathbf{1}\right)$ is $J^{(m_2)}(\mathcal{H}(\mathbf{X}))$. Repeating this for decreasing lengths of $\mathbf{u}$ until $\mathbf{u}$ has length 1 gives $\tilde{\mathbf{X}} = \sigma\left(\mathbf{X}^{(i)} + b^{(i)}\mathbf{1}\right) = A_{L\text{tree}}(\mathbf{X})$. ■

D.1 Assume data is in 1-D

We will refer to a switching set and a rectangular network (defined in Section 3.2 and Section 2.2).

Lemma 17 *Let $m_1, m_2 \in \mathbb{N}, k \in [m_1 m_2]$. A sequence $\{h_i\}$ in $\{-1, 1\}$ that starts with 1 and switches k times is the sum of at most m_2 sequences in $\{-1, 1\}$ that switch at most m_1 times, and the all-ones sequence.*

Proof Suppose h_i switches at $i_1 < \cdots < i_k$. Let $Q = \left\lceil \frac{k}{m_1} \right\rceil \leq m_2$. For $q \in [Q]$, let $\left\{h_i^{(q)}\right\}$ be a sequence in $\{-1, 1\}$ that starts with $(-1)^{q+1}$ and switches precisely at $i \in \{I_1, \cdots, I_k\}$ satisfying $i = j \mod m_2$, which occurs at most m_1 times. Let $s_i = \sum_j h_i^{(q)}$. Then $s_1 = \mathbf{1}\{Q \text{ odd}\} \in \{h_1, h_1 - 1\}$. For $i > 1$,

$$s_i = \begin{cases} s_{i-1} + 2 & \text{if } h_{i-1}^{(q)} = -1, h_i^{(q)} = 1 \text{ for some } q \\ s_{i-1} - 2 & \text{if } h_{i-1}^{(q)} = 1, h_i^{(q)} = -1 \text{ for some } q \\ s_i & \text{else} \end{cases}$$

So $\{s_i\}$ is a sequence in $\{0, -2\}$ or $\{-1, 1\}$ that changes value precisely at $i_1, \ldots, i_k$. Therefore $\{s_i\}$ is either $\{h_i\}$ or $\{h_i - 1\}$. ■

Lemma 18 *Let $p, m \in \mathbf{N}$. Let $\mathbf{z} \in \{-1, 1\}^N$ with at most pm switches. There is an integer $n \leq m$, $\mathbf{w} \in \{-1, 1\}^n$, and a $N \times n$ matrix $\mathbf{H}$ with columns in $\mathbf{H}^{(p)}$ such that $\mathbf{z} = \sigma(\mathbf{H}\mathbf{w})$.*

Proof For $x \in \{-1, 1\}, \sigma(x) = \sigma(x - 1)$. Apply Lemma 17 with $m_2 = p, m_1 = m$. ■

Lemma 19 *$\mathcal{H}(\mathbf{X})$ consists of all columns in $\mathbf{H}^{(1)}$.*

Proof First, $\mathbf{1} = \sigma(\mathbf{0}) = \sigma(X \cdot \mathbf{0}) \in \mathcal{H}(\mathbf{X})$. Next, let $\mathbf{h} \in \mathcal{H}(\mathbf{X}) - \{\mathbf{1}\} \in \{-1, 1\}^N$. By definition of $\mathcal{H}(\mathbf{X})$, there exists $w, b \in \mathbb{R}$ such that $\mathbf{h} = \mathbf{X}w + b\mathbf{1}$. Note $h_1 = 1$ (Remark 14). Let i be the first index at which $\mathbf{h}$ switches. So $x_i w + b < 0 \leq x_{i-1} w + b$, which implies $x_i w < x_{i-1} w$. For all $j > i$, $x_j < x_i$ so $h_j = \sigma(x_j w + b) \leq \sigma(x_i w + b) = \sigma(h_i) = -1$, so $h_j = -1$. So $\mathbf{h}$ switches at most once.

Now, let $\mathbf{h} \in \{-1, 1\}^N$ with $h_1 = 1$. Suppose $\mathbf{h}$ switches once, at index $i \in \{2, \cdots, N\}$. In particular, $h_i = -1$. Let $w = 1, b = -x_{i-1}$. Then at $j < i$, $x_j w + b = x_j - x_{i-1} \geq 0$ so $h_j = \sigma(x_j w + b) = 1$. And for $j \geq i$, $x_j w + b = x_j - x_{i-1} < 0$ so $h_j = -1$. So $\mathbf{h} \in \mathcal{H}(\mathbf{X})$. ■

Proposition 8 *The set $A_{L=3,par}$ contains all columns of $\mathbf{H}^{(m_1)}$.*

Proof Note $A_{L=3,par} = \bigcup_{\mathbf{X}^{(1)}} \mathcal{H}\left(\mathbf{X}^{(\mathbf{1})}\right)$. From Lemma 19, any possible column in $\mathbf{X}^{(\mathbf{1})}$ switches at most once. Apply Lemma 18 with $p = 1, m = m_1$ (so that $mp = m_1$ switches), to any column $\mathbf{z}$ of $\mathbf{X}^{(\mathbf{1})}$. ■

Proposition 9 *For a rectangular network, $A_{L,par}(\mathbf{X})$ is contained in the columns of $\mathbf{H}^{(m_1)}$.*

Proof Let $\mathbf{W}^{(1)} \in \mathbb{R}^{1 \times m_1}$. For any $w, b \in \mathbb{R}$, since the data is ordered, $\sigma\left(\mathbf{X}w + b\mathbf{1}\right) \in \{-1, 1\}^N$ has at most 1 switch. Then $\mathbf{X}^{(1)} = \sigma\left(\mathbf{X}\mathbf{W}^{(1)} + \mathbf{1} \cdot \mathbf{b}^{(1)}\right)$ has m_1 columns each with at most one switch. Therefore $\mathbf{X}^{(1)}$ has at most $m_1 + 1$ unique rows. Let R be the set

of the smallest index of each unique row. We claim that for all layers $l \in [L]$, the rows of $\mathbf{X}^{(l)}$ are constant at indices in $[N] - R$, that is, for all $i \in [N] - R$, the i^{th} and $(i-1)^{\text{th}}$ rows of $\mathbf{X^{(l)}}$ are the same. We prove our claim by induction. The base case for $l = 1$ already holds.

Suppose our claim holds for $l \in \{1, \cdots, L-1\}$. Let $\mathbf{W}^{(l+1)} \in \mathbb{R}^{(m_l \times m_{l+1})}$. The rows of $\mathbf{X}^{(l)}$ are constant at indices in $[N] - R$, so for any $\mathbf{w} \in \mathbb{R}^{m_l}, b \in \mathbb{R}$, the elements of the vector $\mathbf{X}^{(l)}\mathbf{w} + b\mathbf{1}$ are constant at indices in $[N] - R$. This held for any $\mathbf{w} \in \mathbb{R}^{m_l}$, so the rows of $\mathbf{X}^{(l+1)} = \mathbf{X}^{(l)}\mathbf{W}^{(l+1)} + \mathbf{1} \cdot \mathbf{b}^{(l+1)}$ are again constant at indices in $[N] - R$. By induction, our claim holds for all $l \in [L]$. So $\mathbf{X}^{(L)}$ has at most $|R| \leq m_1 + 1$ unique rows and hence has columns that each switch at most m_1 times. ■

Proposition 10 *For a rectangular network, $A_{L,par}(\mathbf{X})$ contains all columns of $\mathbf{H}^{(m_1)}$.*

Proof Let $\mathbf{z} \in \{-1, 1\}^N$ with at most $\min\{N-1, m_1\}$ switches. By Proposition 8, there exists a feasible $\mathbf{X^{(1)}} = \sigma\left(\mathbf{X}\mathbf{W}^{(1)} + \mathbf{1} \cdot \mathbf{b}^{(1)}\right) \in \mathbb{R}^{N \times m}$ and $\mathbf{W^{(2)}} \in \mathbb{R}^{m \times m}$ such that $\mathbf{z}$ is a column of $\mathbf{X^{(2)}} = \sigma\left(\mathbf{X^{(1)}}\mathbf{W^{(2)}} + \mathbf{1} \cdot \mathbf{b}^{(2)}\right)$. Now for every $l \in \{3, \ldots, L-1\}$ we can set $\mathbf{W^{(l)}} = \mathbf{I}_m$ to be the $m \times m$ identity matrix and $\mathbf{b}^{(l)} = \mathbf{0}$ so that $\mathbf{X^{(l)}} = \sigma\left(\mathbf{X^{(l-1)}}\mathbf{W^{(l)}}\right) = \mathbf{X^{(l-1)}}$. Then $\mathbf{X}^{(L)} = \mathbf{X}^{(2)}$ contains $\mathbf{z}$ as a column. Therefore $\mathbf{z} \in A_{L,par}(\mathbf{X})$. ■

Lemma 20 *Let $K = \prod_{l=1}^{L-1} m_l$. The set $A_{L\,tree}(\mathbf{X})$ consists of all columns in $\mathbf{H}^{(K)}$.*

Proof For $l \in [L]$, let $A_l = A_{l\,tree}(\mathbf{X})$. Let $p_k = \prod_{l=1}^{k-1} m_l$. We claim for all $l \in \{2, \cdots, L\}$, A_l consists of all columns in $\mathbf{H}^{(p_l)}$. We prove our claim by induction on l. The base case when $l = 2$ holds by Lemma 19. Now suppose Lemma 20 holds when $l = k \in \{2, \cdots, L-1\}$. Observe $A_k \subset A_{k+1}$, so if $p_k \geq N - 1$, then Lemma 20 holds for $l = k + 1$. So suppose $p_k < N - 1$. Then A_k contains all vectors in $\{-1, 1\}^N$ with at most p_k switches.

Let $\mathbf{h} \in A_{k+1}$. The set A_{k+1} contains all columns in $J^{(m_k)}([A_k])$. So, there exist $\mathbf{w} \in \mathbb{R}^{m_k}, b \in \mathbb{R}$ and a submatrix $\mathbf{Z} = [A_k]_S$ where $|S| = m_k$ and $\mathbf{h} = \sigma(\mathbf{Z}\mathbf{w} + \mathbf{b})$. Each of the at most m_k columns of $\mathbf{Z}$ has at most p_k switches, so the N rows of $\mathbf{Z}$ change at most $m_k p_k = p_{k+1}$ times. So $\mathbf{Z}\mathbf{w} + \mathbf{b}$ changes value, and hence $\mathbf{h} = \sigma(\mathbf{Z}\mathbf{w} + \mathbf{b})$ switches, at most p_{k+1} times. Conversely, by Lemma 18 with $p = p_k, m = m_k$, the set A_{k+1} of all columns in $J^{(m_k)}([A_k])$ contains all vectors with at most $p_k m_k = p_{k+1}$ switches. So our claim holds for $l = k + 1$. By induction, it holds for $l = L$ layers. ■

Proof [Theorem 2] For the parallel network, apply Proposition 7 and then apply Proposition 8 for $L = 3$, and Proposition 9 and Proposition 10 for $L \geq 3$. For tree networks, by Remark 16 the training problem is equivalent to Lasso lower bound with dictionary $A_{L,tree}$ given by Lemma 20. ■

Proof [Lemma 3] By Theorem 2 and Lemma 8, it suffices to show the parameters without unscaling achieve the same objective in the rescaled problem (20) as Lasso. First, $|\mathcal{I}| \leq m_2$ and by Theorem 2, $m^{(i)} \leq m_1$ so the weight matrices are the correct size. Let $S_n^{(i)}$ be the number of times $\mathbf{A}_i$ switches until index n. Since $x_1 > \cdots > x_N$, we

get $\mathbf{X}^{(i,2)}_{n,j} = \sigma\left(\mathbf{X}\mathbf{W}^{(i,1)} + \mathbf{1}\cdot\mathbf{b}^{(i,1)}\right)_{n,j} = \sigma\left(x_n - x_{I^{(i)}_j - 1}\right) = -\sigma\left(S^{(i)}_n - j\right)$. So $\mathbf{X}^{(i,3)}_n = \sigma\left(\mathbf{X}^{(i,2)}\mathbf{W}^{(i,2)} + \mathbf{b}^{(i,2)}\mathbf{1}\right)_n = \sigma\left(\sum_{j=1}^{S^{(i)}_n}(-1)(-1)^{j+1} + \sum_{j=S^{(i)}_n+1}^{m^{(i)}}(1)(-1)^{j+1} - \mathbf{1}\left\{m^{(i)}\text{ odd}\right\}\right) = \sigma\left(-2\cdot\mathbf{1}\left\{S^{(i)}_n\text{ odd}\right\}\right) = (-1)^{S^{(i)}_n} = \mathbf{A}_{n,i}$. So, $f_3(\mathbf{X};\theta) = \xi + \sum_{i\in\mathcal{I}}\alpha_i\mathbf{X}^{(i,3)} = \mathbf{A}\mathbf{z}$. And, $||\boldsymbol{\alpha}||_1 = ||\mathbf{z}^*_I||_1 = ||\mathbf{z}^*||_1$. So the rescaled problem and Lasso achieve the same objective. ■

Remark 17 *For a rectangular network, a reconstruction similar to Lemma 3 holds by setting additional layer weight matrices to the identity.*

Proof [Corollary 1] By Remark 4, $p^*_{L=3,\beta} \le p^*_{L=2,\beta}$. Let $\theta^{(L)}$ and $\boldsymbol{\alpha}^{(L)}$ denote θ and $\boldsymbol{\alpha}$ for a L-layer net. Since the training and rescaled problems have the same optimal value, to show $p^*_{L=2,\beta} \le p^*_{L=3,m_1\beta}$, it suffices to show for any optimal $\theta^{(3)}$, there is $\theta^{(2)}$ with $f_2\left(\mathbf{X};\theta^{(2)}\right) = f_3\left(\mathbf{X};\theta^{(3)}\right)$ and $\left\|\boldsymbol{\alpha}^{(\mathbf{2})}\right\|_1 \le m_1\left\|\boldsymbol{\alpha}^{(\mathbf{3})}\right\|_1$. Let $\mathbf{z}^*$ be optimal in the 3-layer Lasso problem (2). Let $m^*_3 = ||\mathbf{z}^*||_0$ and let $\mathbf{z}\in\mathbb{R}^{m^*_3}$ be the subvector of nonzero elements of $\mathbf{z}^*$. Let $m = m_1 m^*_3$. By Lemma 3 and its proof, there are $\mathbf{W}^{(i,1)}\in\mathbb{R}^{1\times m_1}, \mathbf{b}^{(i,1)}\in\mathbb{R}^{1\times m_1}, \mathbf{W}^{(i,2)}\in\{1,-1,0\}^{m_1}, \mathbf{b}^{(i,2)}\in\mathbb{R}$ such that $\mathbf{X}^{(i,2)}\mathbf{W}^{(i,2)} + \mathbf{b}^{(i,2)}\mathbf{1}\in\{-2,0\}^N$ and $f_3\left(\mathbf{X};\theta\right) =$

$$\sum_{n=1}^{m^*_3} z_i\sigma\left(\mathbf{X}^{(i,2)}\mathbf{W}^{(i,2)} + \mathbf{1}\cdot\mathbf{b}^{(i,2)}\right) = \sum_{i=1}^{m^*_2} z^*_i\left(\mathbf{X}^{(i,2)}\mathbf{W}^{(i,2)} + \mathbf{1}\cdot\mathbf{b}^{(i,2)} + \mathbf{1}\right) =$$

$$\sum_{i=1}^{m^*_3} z^*_i\left(\sigma\left(\mathbf{X}\mathbf{W}^{(i,1)} + \mathbf{1}\cdot\mathbf{b}^{(i,1)}\right)\mathbf{W}^{(i,2)} + \mathbf{1}\cdot\mathbf{b}^{(i,2)} + \mathbf{1}\right) =$$

$$\sigma\left(\mathbf{X}\underbrace{\left[\mathbf{W}^{(1,1)}\cdots\mathbf{W}^{\left(m^*_3,1\right)}\right]}_{\left(\mathbf{W}^{(1,1)},\cdots,\mathbf{W}^{(m,1)}\right)\in\mathbb{R}^{1\times m}} + \mathbf{1}\cdot\underbrace{\left[\mathbf{b}^{(1,1)},\cdots,\mathbf{b}^{\left(m^*_3,1\right)}\right]}_{\left(\mathbf{b}^{(1,1)},\cdots,\mathbf{b}^{(m,1)}\right)\in\mathbb{R}^{1\times m}}\right)\underbrace{\begin{bmatrix} z_1\mathbf{W}^{(1,2)} \\ \vdots \\ z_{m^*_3}\mathbf{W}^{\left(m^*_3,2\right)}\end{bmatrix}}_{\boldsymbol{\alpha}^{(\mathbf{2})}} + \mathbf{1}\underbrace{\sum_{i=1}^{m^*_3} z_i\left(1+\mathbf{b}^{(i,2)}\right)}_{\xi},$$

which is $f_2\left(\mathbf{X};\theta^{(2)}\right)$ with m neurons. And $\|\boldsymbol{\alpha}^{(2)}\|_1 \le m_1\|\mathbf{z}^*\|_1 = m_1\|\boldsymbol{\alpha}^{(3)}\|_1$. ■

D.2 2-D data

Proof [Theorem 3] Lemma 14 holds for any dimension d, so the Lasso formulation in Theorem 1 and Theorem 2 similarly hold for $d > 1$ but with a different dictionary $A_{L,par}$ and matrix $\mathbf{A}$.

Let $x'_n = \angle\mathbf{x}^{(n)}$ and order the data so that $x'_1 > x'_2 > \ldots x'_N$. Let $\mathbf{X}' = (x'_1\cdots x'_N)\in\mathbb{R}^N$. Let $\mathbf{w}\in\mathbb{R}^2$ with $\angle\mathbf{w}\in\left[\frac{\pi}{2},\frac{3\pi}{2}\right]$. Let $w' = \angle\mathbf{w}$. Note $\mathbf{w}\mathbf{x}^{(n)} \ge 0$ if and only if $x'_n \in \left[w'-\frac{\pi}{2}, w'+\frac{\pi}{2}\right]$. Since $x'_n < \pi$ for all $n\in[N]$, this condition is equivalent to $x'_n \ge w'-\frac{\pi}{2}$, ie $n \le \max\left\{n\in[N] : x'_n \ge w'-\frac{\pi}{2}\right\}$. So $\left\{\sigma(\mathbf{X}'\mathbf{w}) : w'\in\left[\frac{\pi}{2},\frac{3\pi}{2}\right]\right\} = \mathbf{H}^{(1)}\cup\{-1\}$. Similarly

$\left\{\sigma(\mathbf{X}'\mathbf{w}) : w' \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]\right\}$ is the "negation" of this set. Therefore, the $L = 2$ training problem is equivalent to the Lasso problem with dictionary $\mathbf{H}^{(1)}$. Proposition 10 holds analogously for this training data, so for $L > 2$, the dictionary is $\mathbf{H}^{(m_{L-1})}$. ■

Proof [Lemma 4] Observe $n \leq i$ if and only if $\mathbf{x}^{(n)}\mathbf{W}^{(i,1)} \geq 0$ and so $\sigma\left(\mathbf{X}\mathbf{W}^{(i,1)}\right) = \left[\mathbf{1}_i^T, -\mathbf{1}_{N-i}^T\right]^T = \mathbf{A}_i$. Thus $f_{L=2}(\mathbf{X};\theta) = \sum_i \alpha_i \sigma\left(\mathbf{X}\mathbf{W}^{(i,1)}\right) = \mathbf{A}\mathbf{z}^*$ so θ achieves the same objective in the rescaled training problem (20), as the optimal value of Lasso (2). Unscaling optimal parameters of the rescaled problem makes them optimal in the training problem. ■

Appendix E. Solution sets of Lasso under minimal regularization

Remark 18 *For each optimal $\mathbf{z}^*$ of the Lasso problem, minimizing the objective over ξ gives the optimal bias term as $\xi^* = \left(\mathbf{y} - \mathbf{1}\mathbf{1}^T\mathbf{y}\right) - \left(\mathbf{A} - \mathbf{1}\mathbf{1}^T\mathbf{A}\right)\mathbf{z}^*$.*

Remark 19 *For a neural net with $L = 2$ layers and sign activation, by Theorem 1 the Lasso problem has an objective function $f(\mathbf{z}) = \frac{1}{2}\|\mathbf{A}\mathbf{z} - \mathbf{y}\|_2^2 + \beta\|\mathbf{z}\|_1$ where $\mathbf{A} \in \mathbb{R}^{N\times N}$. By Lemma 21, $\mathbf{A}$ is full rank, which makes f strongly convex. Therefore the Lasso problem has a unique solution $\mathbf{z}^*$ Boyd and Vandenberghe (2004). Moreover, for any Lasso problem, $\mathbf{z}^*$ satisfies the subgradient condition $\mathbf{0} \in \delta f(\mathbf{z}) = \mathbf{A}^T(\mathbf{A}\mathbf{z}^* - \mathbf{y}) + \beta\partial\|\mathbf{z}^*\|_1$. Equivalently,*

$$\frac{1}{\beta}\mathbf{A}_n^T(\mathbf{A}\mathbf{z}^* - \mathbf{y}) \in \begin{cases} \{-\text{sign}(\mathbf{z}_n^*)\} & \text{if } \mathbf{z}_n^* \neq 0 \\ [-1, 1] & \text{if } \mathbf{z}_n^* = 0 \end{cases}, \quad n \in [N].$$

Proof [Proposition 4] Recall that $\mathbf{e}^{(n)}$ is the n^{th} canonical basis vector as defined in Section 1.2. We analyze the solution set of $\mathbf{A}\mathbf{z}+\xi\mathbf{1} = \mathbf{y}$. We note that $\left(I - \mathbf{1}\mathbf{1}^T/N\right)\mathbf{A}\mathbf{z} = \left(I - \mathbf{1}\mathbf{1}^T/N\right)\mathbf{y}$.. As $\mathbf{z}^*$ is optimal in (9) , this implies that $(I - \mathbf{1}\mathbf{1}^T/N)\mathbf{A}\mathbf{z}^* = (I - \mathbf{1}\mathbf{1}^T/N)\mathbf{y}$. This implies that $(I - \mathbf{1}\mathbf{1}^T/N)\mathbf{A}(\mathbf{z} - \mathbf{z}^*) = 0$. As $x_1 > x_2 > \cdots > x_N$, we have $\mathbf{A}\left(\mathbf{e}^{(1)} + \mathbf{e}^{(N)}\right) \propto \mathbf{1}$. As $\mathbf{A}$ is invertible by Lemma 22 in Appendix F, this implies that there exists $t \in \mathbb{R}$ such that $\mathbf{z} - \mathbf{z}^* = t\left(\mathbf{e}^{(1)} + \mathbf{e}^{(N)}\right)$. It is impossible to have $z_1^* z_N^* > 0$ from the optimality of $\mathbf{z}^*$. Otherwise, by taking $t = -\text{sign}(z_1^*)\min\{|z_1^*|, |z_n^*|\}$, we have $\|\mathbf{z}\|_1 = \|\mathbf{z}^*\|_1 - 2\min\{|z_1^*|, |z_n^*|\} < \|\mathbf{z}^*\|_1$. Therefore, we have $z_1^* z_n^* \leq 0$. We can reparameterize $\mathbf{z} = \mathbf{z}^* + t\text{sign}(z_1^*)(\mathbf{e}^{(1)} + \mathbf{e}^{(N)})$. It is easy to verify that for t such that $-|z_1^*| \leq t \leq |z_n^*|$, we have $\|\mathbf{z}\|_1 = \|\mathbf{z}_n\|_1$, while for other choice of t, we have $\|\mathbf{z}\|_1 > \|\mathbf{z}_n\|_1$. Therefore, the solution set of (9) is given by (10). ■

Proof [Proposition 5] Follows from Remark 19 describing the Lasso objective. ■

Proof [Lemma 2] The result follows from the definition of $\mathbf{A}$, the assumption that the data is ordered and that the output of sign activation is σ is ± 1 for sign activation. ■

Proof [Proposition 6] By Lemma 23, for $n \in [N-1]$, $z_{+n}^* = y_n - y_{n-1} \geq 0$ and $z_{+N}^* = y_N \geq 0$. So $\mathbf{z}^*$ achieves an objective value of $\|\mathbf{z}^*\|_1 = y_1$ in (9). Now let $\mathbf{z}$ be any solution to (9). Then $\mathbf{A}\mathbf{z} = \mathbf{y}$. Since the first row of $\mathbf{A}$ is $[\mathbf{1}^T, \mathbf{0}^T]$, we have

$y_1 = (\mathbf{Az})_1 = \mathbf{1}^T \mathbf{z}_+ \leq \|\mathbf{z}_+\| \leq \|\mathbf{z}\|_1 \leq \|\mathbf{z}^*\| = y_1$. So $\|\mathbf{z}_+\|_1 = \|\mathbf{z}\|_1 = y_1$, leaving $\mathbf{z}_- = \mathbf{0} = \mathbf{z}_-^*$. Therefore $\mathbf{z}_+ = \mathbf{A}^{-1}\mathbf{y} = \mathbf{z}_+^*$. Applying Lemma 23 gives the result. ■

Proof [Corollary 2] By Lemma 21, Lemma 22, Lemma 23 and Lemma 24, the dictionary matrix for the 2-layer net is full rank for sign, absolute value, threshold and ReLU activations. The dictionary matrices for deeper nets with sign activation are also full rank by Remark 4. Let $\mathbf{u} = \mathbf{A}^T(\mathbf{Az}^* - \mathbf{y})$. By Remark 19, as $\beta \to 0$, we have $\mathbf{u} \to \mathbf{0}$, so $\mathbf{Az} - \mathbf{y} = (\mathbf{AA}^T)^{-1}\mathbf{Au} \to \mathbf{0}$. So as $\beta \to 0$, the optimal Lasso objective approaches 0, and by Theorem 2 and Theorem 1, so does the training problem. So $f_L(\mathbf{X};\theta) \xrightarrow{\beta \to 0} \mathbf{y}$. ■

Proof [Lemma 5] Every ReLU deep narrow feature $f_{\mathbf{s},\mathbf{j},k}^{(L,\sigma)}(\mathbf{X})$ has slope of magnitude at most 1. Also, $\mathbf{y} = f_L(\mathbf{X};\theta) = \xi^*\mathbf{1} + \mathbf{Az}^* = \xi^*\mathbf{1} + \sum_i z_i^* \mathbf{A}_i$. For any $n \in [N-1]$, $|\mu_n| = \left|\frac{f_L(\mathbf{X};\theta)_{n+1} - f_L(\mathbf{X};\theta)_n}{x_{n+1} - x_n}\right| = \left|\frac{\sum_i z_i^*(\mathbf{A}_{n+1,i} - \mathbf{A}_{n,i})}{x_{n+1} - x_n}\right| = \left|\sum_i z_i^* \frac{f_{\mathbf{s},\mathbf{j},k}^{(L,\sigma)}(x_{n+1}) - f_{\mathbf{s},\mathbf{j},k}^{(L,\sigma)}(x_n)}{x_{n+1} - x_n}\right|$
$\leq \sum_i |z_i^*| \left|\frac{f_{\mathbf{s},\mathbf{j},k}^{(L,\sigma)}(x_{n+1}) - f_{\mathbf{s},\mathbf{j},k}^{(L,\sigma)}(x_n)}{x_{n+1} - x_n}\right| \leq \sum_i |z_i^*| = \|\mathbf{z}^*\|$. ■

Proof [Lemma 6] Let $n \in [N-1]$. Let $\mathcal{S}_n^+ = \{i \in [N] : i > n, (z_+)_i \neq 0\}, \mathcal{S}_n^- = \{i \in [N] : i \leq n, (z_-)_i \neq 0\}$. Observe $\mathcal{S}_{n+1}^+ = \mathcal{S}_n^+ - \{n+1\}$ if $(z_+)_{n+1} \neq 0$ and $\mathcal{S}_{n+1}^+ = \mathcal{S}_n^+$ otherwise. Similarly, $\mathcal{S}_{n+1}^- = \mathcal{S}_n^+ \cup \{n+1\}$ if $(z_-)_{n+1} \neq 0$ and $\mathcal{S}_{n+1}^- = \mathcal{S}_n^-$ otherwise. Now, $\mathrm{ReLU}_{x_i}^+$ has slope 1 after x_i and $\mathrm{ReLU}_{x_i}^-$ has slope 1 before x_i, so $\mu_n = \sum_{i \in \mathcal{S}_n^+} (z_+)_i + \sum_{i \in \mathcal{S}_n^-} (z_-)_i$. Combining this with the previous observation, $\mu_n - \mu_{n+1} = -(z_+)_{n+1} + (z_-)_{n+1}$. Note we used $\mu_N = 0$. So $|\mu_n - \mu_{n+1}| = \left|-(z_+)_{n+1} + (z_-)_{n+1}\right| \leq \left|(z_+)_{n+1}\right| + \left|(z_-)_{n+1}\right|$. So $\sum_{n=1}^{N-1} |\mu_n - \mu_{n+1}| \leq \|\mathbf{z}^*\|_1 - |(z_+)_1| - |(z_-)_1| \leq \|\mathbf{z}^*\|_1$.

Now, for any $n \in [N-1]$, $(\mathbf{Az} + \xi\mathbf{1})_n - (\mathbf{Az} + \xi\mathbf{1})_{n+1} = \sum_{i=1}^N (z_+)_i \left(\mathbf{A}_{+n,i} - \mathbf{A}_{+n+1,i}\right) = \sum_{i=1}^N (z_+)_i \left((x_n - x_i)_+ - (x_{n+1} - x_i)_+\right) = \sum_{i=n+1}^N (\mu_{i-1} - \mu_i)(x_n - x_{n+1}) = (x_n - x_{n+1})\mu_n = y_n - y_{n+1}$. And $(\mathbf{Az} + \xi\mathbf{1})_N = \xi + \sum_{i=1}^N (z_+)_i (x_N - x_i)_+ = y_N + \sum_{i=1}^N (z_+)_i (0) = y_N$. So $\mathbf{Az} + \xi\mathbf{1} = \mathbf{y}$. And $\|\mathbf{z}\|_1 = \sum_{n=1}^{N-1} |\mu_n - \mu_{n+1}|$, which exactly hits the lower bound on $\|\mathbf{z}^*\|_1$. Therefore $\mathbf{z}, \xi$ is optimal. ■

Remark 20 *By Lemma 22, the absolute value network "de-biases" the target vector, normalizes is by the interval lengths $x_i - x_{i+1}$, and applies $\mathbf{E}$ (which contains the difference matrix Δ) twice, acting as a second-order difference detector. By Lemma 21, the sign network's dictionary inverse contains Δ just once, acting as a first-order difference detector.*

Appendix F. Inverses of 2-layer dictionary matrices

In this section, we consider the 2-layer dictionary matrix $\mathbf{A}$ as defined in Remark 2. Define the *finite difference matrix*

$$\Delta = \begin{pmatrix} 1 & -1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & -1 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -1 \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix} \in \mathbb{R}^{N-1 \times N-1}. \tag{39}$$

Multiplying a matrix on its right by Δ subtracts its consecutive rows. Define the diagonal matrix $\mathbf{D} \in \mathbb{R}^{N \times N}$ by $\mathbf{D}_{i,i} = \frac{1}{x_i - x_{i+1}}$ for $i \in [N-1]$ and $\mathbf{D}_{N,N} = \frac{1}{x_1 - x_N}$. For $n \in \mathbb{N}$, let

$$\mathbf{A}_n^{(s)} = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ -1 & 1 & 1 & \cdots & 1 \\ -1 & -1 & 1 & \cdots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & -1 & \cdots & 1 \end{pmatrix}, \mathbf{A}_n^{(t)} = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 0 & 1 & 1 & \cdots & 1 \\ 0 & 0 & 1 & \cdots & 1 \\ \vdots & \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix} \in \mathbb{R}^{n \times n}. \tag{40}$$

Remark 21 *The dictionary matrices for sign and threshold activation satisfy* $\mathbf{A} = \mathbf{A}_N^{(s)}$ *and* $\mathbf{A}_+ = \mathbf{A}_N^{(t)}$*, respectively.*

Lemma 21 *The dictionary matrix for sign activation has inverse*

$$\mathbf{A}^{-1} = \tfrac{1}{2} \left(\begin{array}{cccc|c} & & & & 0 \\ & & \Delta & & \vdots \\ & & & & 0 \\ & & & & -1 \\ \hline 1 & 0 & \cdots & 0 & 1 \end{array}\right).$$

Proof Multiplying the two matrices (see Remark 21) gives the identity. ■

Lemma 22 *The dictionary matrix* $\mathbf{A}$ *for absolute value activation has inverse* $\mathbf{A}^{-1} = \frac{1}{2}\mathbf{PEDE}$*, where* $\mathbf{P} = \left(\begin{array}{cccc|c} 0 & 0 & \cdots & 0 & 1 \\ \hline & & & & 0 \\ & & I_{N-1} & & \vdots \\ & & & & 0 \\ & & & & 0 \end{array}\right), \mathbf{E} = \left(\begin{array}{cccc|c} & & & & 0 \\ & & \Delta & & \vdots \\ & & & & 0 \\ & & & & -1 \\ \hline -1 & 0 & \cdots & 0 & -1 \end{array}\right).$

Proof For $i \in [N-1], j \in [N]$, $\mathbf{A}_{i,j} - \mathbf{A}_{i+1,j} = \begin{cases} (x_j - x_i) - (x_j - x_{i+1}) = x_{i+1} - x_i & \text{if } i > j \\ (x_i - x_j) - (x_{i+1} - x_j) = x_i - x_{i+1} & \text{if } i \leq j \end{cases}$.
And for all $j \in [N]$, $\mathbf{A}_{1,j} + \mathbf{A}_{N,j} = (x_1 - x_j) + (x_j - x_N) = x_1 + x_N$. Therefore,

$$\mathbf{DEA} = \left(\begin{array}{c|ccc} -1 & & & \\ \vdots & & \mathbf{A}_{N-1}^{(s)} & \\ -1 & & & \\ \hline -1 & -1 & \cdots & -1 \end{array}\right), \quad \frac{1}{2}\mathbf{EDEA} = \left(\begin{array}{c|ccc} 0 & & & \\ \vdots & & I_{N-1} & \\ 0 & & & \\ \hline 1 & 0 & \cdots & 0 \end{array}\right).$$

Applying the permutation $\mathbf{P}$ makes $\frac{1}{2}\mathbf{PEDEA} = \mathbf{I}$, so $\mathbf{A^{-1}} = \frac{1}{2}\mathbf{PEDE}$. ■

Lemma 23 *The inverse of* $\mathbf{A}_+$ *for threshold activation is* $\mathbf{A}_+^{-1} = \left(\begin{array}{ccc|c} & \Delta & & \begin{matrix}0\\\vdots\\0\\-1\end{matrix} \\ \hline 0 & 0 \cdots & 0 & 1 \end{array}\right)$.

Proof Multiplying the two matrices (see Remark 21) gives the identity. ■

Lemma 24 *The submatrix* $[(\mathbf{A}_+)_{1:N,2:N}, (\mathbf{A}_-)_{1:N,1}] \in \mathbb{R}^{N\times N}$ *of the dictionary matrix for ReLU activation has inverse* $\mathbf{E}_+\mathbf{DE}_-$*, where*

$$\mathbf{E}_+ = \left(\begin{array}{ccc|c} & \Delta & & \begin{matrix}0\\\vdots\\0\\1\end{matrix} \\ \hline 0 & 0 \cdots & 0 & 1 \end{array}\right), \quad \mathbf{E}_- = \left(\begin{array}{ccc|c} & \Delta & & \begin{matrix}0\\\vdots\\0\\-1\end{matrix} \\ \hline 0 & 0 \cdots & 0 & -1 \end{array}\right).$$

Proof For $i \in [N-1], j \in [N]$,

$$(\mathbf{A}_+)_{i,j} - (\mathbf{A}_+)_{i+1,j} = \begin{cases} 0 & \text{if } i \geq j \\ (x_i - x_j) - (x_{i+1} - x_j) = x_i - x_{i+1} & \text{if } i < j \end{cases}$$

and $(\mathbf{A}_-)_{i,1} - (\mathbf{A}_-)_{i+1,1} = (x_1 - x_i) - (x_1 - x_{i+1}) = x_{i+1} - x_i$. Observe that $\mathbf{DE}_-\mathbf{A} = \left(\begin{array}{ccc|c} & \mathbf{A}_{N-1}^{(t)} & & \begin{matrix}-1\\\vdots\\-1\\-1\end{matrix} \\ \hline 0 & 0 \cdots & 0 & 1 \end{array}\right)$, and applying $\mathbf{E}_+$ gives $\mathbf{I}$. ■

Appendix G. Solution path for sign activation and binary, periodic labels

In this section we assume the neural net uses sign activation, and $d = 1$. We wil refer to $\mathbf{e}^{(n)}$ as the n^{th} canonical basis vector, as defined in Section 1.2.

Remark 22 *A neural net with all weights being* 0 *achieves the same objective in the training problem as the optimal Lasso value and is therefore optimal.*

We will find β_c. Then for $\beta < \beta_c$, we use the subgradient condition from Remark 19 to solve the Lasso problem (2). Note when $L = 2$, $(\mathbf{A}_n)^T = (\mathbf{1}_n, -\mathbf{1}_{N-n})$ switches at $n + 1$.

G.1 Assume $L = 2$

Lemma 25 *The elements of* $\mathbf{A}^T\mathbf{A} \in \mathbb{R}^{N\times N}$ *are* $(\mathbf{A}^T\mathbf{A})_{i,j} = N - 2|i-j|$.

Proof If $1 \le i \le j \le N$ then $(\mathbf{A}^T\mathbf{A})_{i,j} = \sum_{k=1}^{i-1}\mathbf{A}_{i,k}\mathbf{A}_{j,k} + \sum_{k=i}^{j-1}\mathbf{A}_{i,k}\mathbf{A}_{j,k} + \sum_{k=j}^{N}\mathbf{A}_{i,k}\mathbf{A}_{j,k} = \sum_{k=1}^{i-1}(1)(1) + \sum_{k=i}^{j-1}(-1)(1) + \sum_{k=j}^{N}(-1)(-1) = (i-1) - (j-1-i+1) + (N-j+1) = N + 2(i-j) = N - 2|i-j|$. Since $\mathbf{A}^T\mathbf{A}$ is symmetric, if $1 \le j \le i \le N$ then $(\mathbf{A}^T\mathbf{A})_{i,j} = (\mathbf{A}^T\mathbf{A})_{j,i} = N + 2(j-i) = N - 2|i-j|$. So for any $i, j \in [N]$, $(\mathbf{A}^T\mathbf{A})_{i,j} = N - 2|i-j|$. ■

Remark 23 *By Lemma 25,* $\mathbf{A}^T\mathbf{A}$ *is of the form*

$$\mathbf{A}^T\mathbf{A} = \begin{pmatrix} N & N-2 & N-4 & \cdots & 2 & 0 & -2 & \cdots & 6-N & 4-N & 2-N \\ N-2 & N & N-2 & \cdots & 4 & 2 & 0 & \cdots & 8-N & 6-N & 4-N \\ N-4 & N-2 & N & \cdots & 6 & 4 & 2 & \cdots & 10-N & 8-N & 6-N \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 6-N & 8-N & 10-N & \cdots & 2 & 4 & 6 & \cdots & 10-N & 8-N & 6-N \\ 4-N & 6-N & 8-N & \cdots & 0 & 2 & 4 & \cdots & 8-N & 6-N & 4-N \\ 2-N & 4-N & 6-N & \cdots & -2 & 0 & 2 & \cdots & N-4 & N-2 & N \end{pmatrix}. \tag{41}$$

An example of a column of $\mathbf{A}^T\mathbf{A}$ is plotted in the left plot of Figure 15.

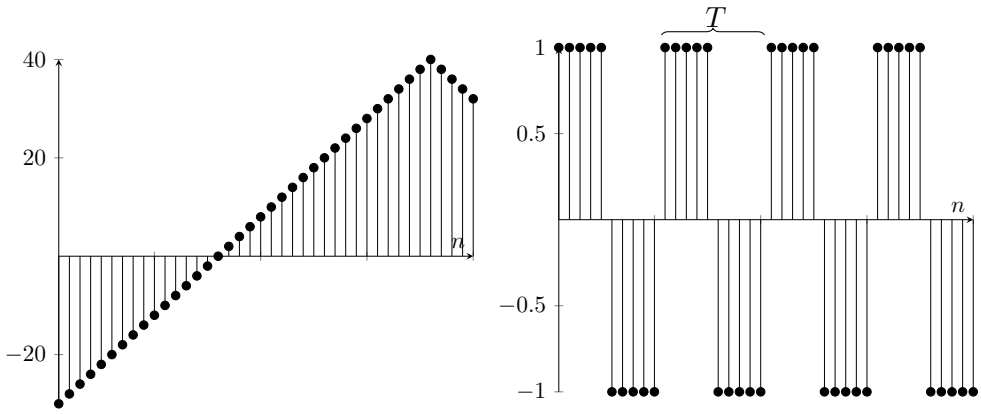


Figure 15: Left: Column 10 of $\mathbf{A}^T\mathbf{A}$ for $N = 40$. Right: Vector $\mathbf{h}^{(T)}$ for $T = 10$.

Remark 24 *For* $n \in [N]$, $(\mathbf{A}^T\mathbf{y})_n = \sum_{i=1}^{n}\mathbf{y}_i - \sum_{i=n+1}^{N}\mathbf{y}_i$.

Definition 13 For $a, b \in \mathbb{Z}$, denote $\mathrm{Quot}(a,b) \in \mathbb{Z}$ and $\mathrm{Rem}(a,b) \in \{0, \cdots, b-1\}$ as the quotient and remainder, respectively, when a is divided by b. Define the *modified remainder*

$$\mathrm{rem}(a,b) = \begin{cases} \mathrm{Rem}(a,b) & \text{if } \mathrm{Rem}(a,b) > 0 \\ b & \text{if } \mathrm{Rem}(a,b) = 0 \end{cases} \in [b].$$

Next define the *modified quotient*

$$\text{quot}(a,b) = \frac{a - \text{rem}(a,b)}{b} = \begin{cases} \text{Quot}(a,b) & \text{if Rem}(a,b) > 0 \\ \text{Quot}(a,b) - 1 & \text{if Rem}(a,b) = 0 \end{cases}.$$

.

Remark 25 *The square wave has elements* $\mathbf{h}_n^{(T)} = \begin{cases} -1 & \text{if} \quad rem(T,n) \leq T/2 \\ 1 & \text{else.} \end{cases}$

Remark 26 *Since* $\mathbf{h}^{(T)}$ *is periodic and zero mean, for* $i, n \geq 0$, $\sum_{j=iT+1}^{nT} \mathbf{h}^{(T)}{}_j = 0$.

Lemma 26 *The vector* $\mathbf{A}^T\mathbf{h}^{(T)}$ *is periodic with period* T. *For* $n \in [T]$,

$$\left(\mathbf{A}^T\mathbf{h}^{(T)}\right)_n = 2\begin{cases} n & \text{if } n \leq \frac{T}{2} \\ T-n & \text{else} \end{cases} \quad \in [0,T].$$

Proof By Remark 24, Remark 26, and periodicity of $\mathbf{h}^{(T)}$, for $n \in [T], j \in [2k-1]$,

$$\left(\mathbf{A}^T\mathbf{h}^{(T)}\right)_{n+jT} = \sum_{i=jT+1}^{jT+n} \mathbf{h}^{(T)}{}_i - \sum_{i=n+jT+1}^{(j+1)T} \mathbf{h}^{(T)}{}_i = \begin{cases} \sum_{i=1}^{n} 1 - \sum_{i=n+1}^{T/2} 1 + \sum_{i=1+T/2}^{T} 1 & \text{if } n \leq \frac{T}{2} \\ \sum_{i=1}^{T/2} 1 - \sum_{i=\frac{T}{2}+1}^{n} 1 + \sum_{i=n+1}^{T} 1 & \text{else.} \end{cases}$$

Simplifying gives the result. ■

Lemma 27 *Let* $q_n = \text{quot}\left(n, \frac{T}{2}\right) \in \{0, \cdots, 2k-1\}$. *Then*

$$\left(\mathbf{A}^T\mathbf{h}^{(T)}\right)_n = 2(-1)^{q_n+1}\text{rem}\left(n, \frac{T}{2}\right) - \mathbf{1}\{q_n \ odd\}T. \tag{42}$$

Proof Follows from Lemma 26. ■

Corollary 5 *Suppose* $\mathbf{z} = \mathbf{e}^{(\frac{T}{2})} + \mathbf{e}^{\left(N-\frac{T}{2}\right)}$. *Then for* $n \leq \frac{T}{2}$ *and* $n \geq N - \frac{T}{2}$, $\frac{1}{2}(\mathbf{A}^T\mathbf{A}\mathbf{z})_n = \left(\mathbf{A}^T\mathbf{h}^{(T)}\right)_n$. *And if* $\frac{T}{2} \leq n \leq N - \frac{T}{2}$, *then* $(\mathbf{A}^T\mathbf{A}\mathbf{z})_n = 2T$.

Proof By Lemma 25, for $n \in [N]$, $(\mathbf{A}^T\mathbf{A}\mathbf{z})_n = 2\left(N - \left|n - \frac{T}{2}\right| - \left|n - N + \frac{T}{2}\right|\right)$
Simplifying and applying Lemma 26 gives the result. ■

Lemma 28 *If* $\mathbf{y} = \mathbf{h}^{(T)}$, *then the critical* β *(defined in Section 6) is* $\beta_c = T$.

Proof By Remark 22, $\beta_c = \max_{n\in[N]} |\mathbf{A}_n^T \mathbf{y}| = \max_{n\in[N]} \left| \left(\mathbf{A}^T \mathbf{h}^{(T)} \right)_n \right| = T.$ ■

Lemma 29 *Let* $\mathbf{y} = \mathbf{h}^{(T)}$. *If* $\tilde{\beta} \geq \frac{1}{2}$ *then the solution to the Lasso problem* (2) *is* $\mathbf{z}^* = \frac{1}{2}\left(1-\tilde{\beta}\right)_+ \left(\mathbf{e}^{\left(\frac{T}{2}\right)} + \mathbf{e}^{\left(N-\frac{T}{2}\right)}\right)$.

Proof By Lemma 28, $\beta_c = T$. By Lemma 28, if $\tilde{\beta} \geq 1$ then $\mathbf{z}^* = \mathbf{0}$ as desired. Now suppose $\frac{1}{2} \leq \tilde{\beta} \leq 1$. Let $\delta = 1-\tilde{\beta}, \mathbf{g} = \mathbf{A}_n^T (\mathbf{A}\mathbf{z}^* - \mathbf{y})$. By Corollary 5 and Lemma 26,

$$\mathbf{g} = \begin{cases} (\delta-1)\left(\mathbf{A}^T\mathbf{h}^{(T)}\right)_n = -2\tilde{\beta}n & \in [-\beta, 0] \quad \text{if } n \leq \dfrac{T}{2} \\ \left(\delta T - \left(\mathbf{A}^T\mathbf{h}_{\mathbf{k},\mathbf{T}}\right)_n\right) = \left(\beta_c - \beta - (\mathbf{A}^T\mathbf{h}^{(T)})_n\right) & \in [-\beta, \beta] \quad \text{if } \dfrac{T}{2} \leq n \leq N - \dfrac{T}{2} \\ (\delta-1)\left(\mathbf{A}^T\mathbf{h}^{(T)}\right)_n = -2\tilde{\beta}(N-n) & \in [-\beta, 0] \quad \text{if } N - \dfrac{T}{2} \leq n \end{cases},$$

where the second set inclusion follows from $\left(\mathbf{A}^T\mathbf{h}^{(T)}\right)_n \in [0, \beta_c]$ by Lemma 26 so that $-\beta \leq \mathbf{g} \leq \beta_c - \beta \leq \beta$. Therefore, $|\mathbf{A}_n^T (\mathbf{A}\mathbf{z}^* - \mathbf{y})| \leq \beta$, and at $n \in \{n : \mathbf{z}_n^* \neq 0\} = \left\{\frac{T}{2}, N - \frac{T}{2}\right\}$, we have $\mathbf{A}_n^T (\mathbf{A}\mathbf{z}^* - \mathbf{y}) = -\tilde{\beta}\frac{T}{2} = -\beta = -\beta \text{sign}(\mathbf{z}_n^*)$. By Remark 19, $\mathbf{z}^*$ is optimal. ■

Lemma 30 *Let* $a, b, c, d \in \mathbf{Z}_+, d \in \mathbb{R}, r = 1 - \text{rem}(b-a, 2)$. *Then,*

$$\begin{aligned} \sum_{j=a}^{b} (-1)^j (c - jd) &= (-1)^a \left((c-ad)r + (-1)^r \frac{(b-(a+r)+1)d}{2} \right) \\ &= (-1)^a \begin{cases} \frac{(b-a+1)d}{2} & \text{if } r = 0 \\ c - ad - \frac{(b-a)d}{2} & \text{else} \end{cases}. \end{aligned} \tag{43}$$

Proof We have

$$\begin{aligned} \sum_{j=a}^{b} (-1)^j (c-jd) &= (-1)^a (c-ad)r + \sum_{j=a+r}^{b} (-1)^j (c-jd) \\ &= (-1)^a (c-ad)r + \sum_{\substack{a+r \leq j \leq b-1 \\ j-(a+r) \text{ is even}}} (-1)^j (c-jd) + (-1)^{j+1} (c-(j+1)d) \\ &= (-1)^a (c-ad)r + (-1)^{a+r} \sum_{\substack{a+r \leq j \leq b-1 \\ j-(a+r) \text{ is even}}} (c-jd) - (c-(j+1)d) \\ &= (-1)^a (c-ad)r + (-1)^{a+r} \sum_{\substack{a+r \leq j \leq b-1 \\ j-(a+r) \text{ is even}}} d \\ &= (-1)^a \left((c-ad)r + (-1)^r \frac{b-(a+r)+1}{2} d \right). \end{aligned}$$

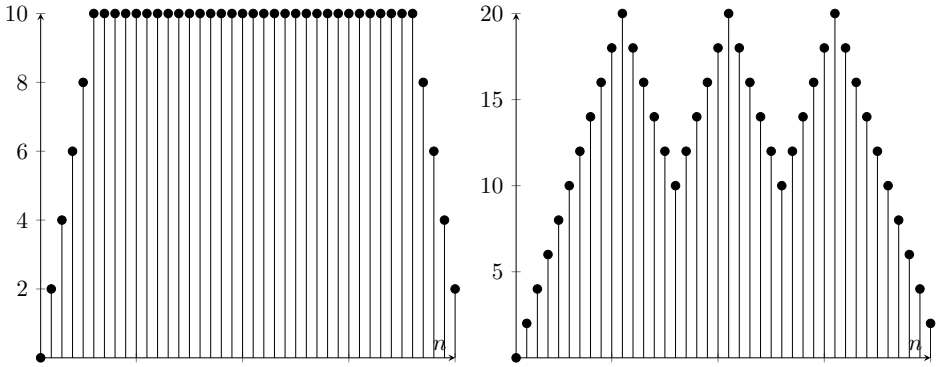


Figure 16: Examples of $\frac{1}{2w_{bdry}}\mathbf{A}^T\mathbf{A}\mathbf{z}_{\mathbf{bdry}}$ (left) and $\frac{1}{w_{cycle}}\mathbf{A}^T\mathbf{A}\mathbf{z}_{\mathbf{cycle}}$ (right).

Simplifying gives (43). ■

Lemma 31 *Consider a $L = 2$ layer neural network. Suppose $\mathbf{y} = \mathbf{h}^{(T)}$ and $0 < \beta < \frac{\beta_c}{2}$. Let $w_{bdry} = 1 - \frac{3}{2}\tilde{\beta}, w_{cycle} = 2\tilde{\beta} - 1$. Let $\mathbf{z}_{\mathbf{bdry}} = w_{bdry}\left(\mathbf{e}^{\left(\frac{T}{2}\right)} + \mathbf{e}^{\left(N-\frac{T}{2}\right)}\right), \mathbf{z}_{\mathbf{cycle}} = w_{cycle}\sum_{i=2}^{2k-2}(-1)^i\mathbf{e}^{\left(\frac{T}{2}i\right)} \in \mathbb{R}^N$. Then $\mathbf{z}^* = \mathbf{z}_{\mathbf{bdry}} + \mathbf{z}_{\mathbf{cycle}}$ solves the Lasso problem (2).*

Remark 27 *The nonzero indices of $\mathbf{z}_{\mathbf{bdry}}$ and $\mathbf{z}_{\mathbf{cycle}}$ partition those that are multiples of $\frac{T}{2}$.*

Proof We show $\mathbf{z}^*$ is optimal using the subgradient condition in Remark 19. By Corollary 5,

$$\frac{1}{2w_{bdry}}(\mathbf{A}^T\mathbf{A}\mathbf{z}_{\mathbf{bdry}})_n = \begin{cases} \left(\mathbf{A}^T\mathbf{h}^{(T)}\right)_n & \text{if } n \le \frac{T}{2} \text{ or } n \ge N - \frac{T}{2} \\ T & \text{if } \frac{T}{2} \le n \le N - \frac{T}{2} \end{cases}, \quad n \in [N]. \tag{44}$$

Next, $\frac{1}{w_{cycle}}(\mathbf{A}^T\mathbf{A}\mathbf{z}_{\mathbf{cycle}})_n = \sum_{j=2}^{2k-2}(-1)^j\left(N - 2\left|n - j\frac{T}{2}\right|\right)$. Expanding, simplifying and comparing with Lemma 26 gives the following.

First suppose $n \le \frac{T}{2}$ or $n \ge N - \frac{T}{2}$. Then, $\frac{1}{w_{cycle}}(\mathbf{A}^T\mathbf{A}\mathbf{z}_{\mathbf{cycle}})_n = \sum_{j=2}^{2k-2}(-1)^j\left(N + 2s\left(n - j\frac{T}{2}\right)\right)$, where $s = 1$ if $n \le \frac{T}{2}$ and $s = -1$ if $n \ge N - \frac{T}{2}$. Applying Lemma 30 with $a = 2, b = 2k-2, c = N + 2sn, d = sT, r = 1 - \text{rem}(b-a, 2) = 1$ gives $(\mathbf{A}^T\mathbf{A}\mathbf{z}_{\mathbf{cycle}})_n = w_{cycle}(-1)^a\left(c - ad - \frac{(b-a)d}{2}\right) = w_{cycle}\left(N + 2sn - 2sT - s(k-2)T\right) = w_{cycle}\left(N - skT + 2sn\right)$. Plugging in s and $N = kT$ and applying Lemma 26 gives

$$\frac{1}{w_{cycle}}(\mathbf{A}^T\mathbf{A}\mathbf{z}_{\mathbf{cycle}})_n = 2\begin{cases} n & \text{if } n \le \frac{T}{2} \\ N - n & \text{if } n \ge N - \frac{T}{2} \end{cases} = (\mathbf{A}^T\mathbf{h}^{(T)})_n. \tag{45}$$

Next suppose $\frac{T}{2} \le n \le N - \frac{T}{2}$. Let $q_n = \text{quot}\left(n, \frac{T}{2}\right), r_n = \text{rem}\left(n, \frac{T}{2}\right)$. Then

$$\frac{1}{w_{cycle}}(\mathbf{A}^T\mathbf{A}\mathbf{z}_{\mathbf{cycle}})_n = \sum_{j=2}^{q_n}(-1)^j\left(N - 2\left(n - j\frac{T}{2}\right)\right) + \sum_{j=q_n+1}^{2k-2}(-1)^j\left(N + 2\left(n - j\frac{T}{2}\right)\right).$$

Let $a_+ = 2, b_+ = q_n, c_+ = N - 2n, d_+ = -T, a_- = q_n + 1 = b_+ + 1, b_- = 2k - 2, c_- = N + 2n, d_- = T = -d_+, r_+ = \mathbf{1}\{b_+ - a_+ \text{ even}\} = \mathbf{1}\{b_+ \text{ even}\}, r_- = \mathbf{1}\{b_- - a_- \text{ even}\} = \mathbf{1}\{b_+ \text{ odd}\}$. Note that $(-1)^{\mathbf{1}\{b_+ \text{ even}\}} = (-1)^{b+1}, (-1)^{\mathbf{1}\{b_+ \text{ odd}\}} = (-1)^b$, and $n - \frac{T}{2}q_n = \text{Rem}\left(n, \frac{T}{2}\right)$. By Lemma 30, $\frac{1}{w_{cycle}}\left(\mathbf{A}^T\mathbf{A}\mathbf{z_{cycle}}\right)_n =$

$$
\begin{aligned}
&(-1)^{a_+}\left((c_+ - a_+d_+)\mathbf{1}\{b_+ \text{ even }\} + (-1)^{1+b_+}\frac{b_+ - (a_+ + \mathbf{1}\{b_+ \text{ even }\}) + 1}{2}d_+\right)\\
&+ (-1)^{1+b_+}\left((c_- + (1+b_+)d_+)\mathbf{1}\{b_+ \text{ odd }\} + (-1)^{1+b_+}\frac{b_- - (1+b_+ + \mathbf{1}\{b_+ \text{ odd }\}) + 1}{2}d_+\right)\\
&= \begin{cases} N - 2n + 2T + \frac{q_n - 2}{2}T - \frac{2k-2-q_n}{2}T = 2\left(T - \text{rem}\left(n, \frac{T}{2}\right)\right) & \text{if } b_+ = q_n \text{ is even} \\ \frac{1-q_n}{2}T + N + 2n - (1+q_n)T - \frac{2k-3-q_n}{2}T = T + 2\text{rem}\left(n, \frac{T}{2}\right) & \text{if } b_+ = q_n \text{ is odd} \end{cases}\\
&= (2 - \mathbf{1}\{q_n \text{ odd}\})T + 2(-1)^{q_n+1}\text{rem}\left(n, \frac{T}{2}\right)\\
&= 2T - \left(\mathbf{A}^T\mathbf{h}_{\mathbf{k},\mathbf{T}}\right)_n,
\end{aligned}
$$

where the last equality follows from Lemma 27. Combining with (45) gives

$$
(\mathbf{A}^T\mathbf{A}\mathbf{z_{cycle}})_n = w_{cycle} \cdot \begin{cases} \left(\mathbf{A}^T\mathbf{h}^{(T)}\right)_n & \text{if } n \le \frac{T}{2} \text{ or } n \ge N - \frac{T}{2} \\ 2T - \left(\mathbf{A}^T\mathbf{h}^{(T)}\right)_n & \text{if } \frac{T}{2} \le n \le N - \frac{N}{2}. \end{cases} \tag{46}
$$

Add (44) and (46) and plug in $\mathbf{y} = \mathbf{h}^{(T)}$ to get

$$
(\mathbf{A}^T\mathbf{A}\mathbf{z})_n = \begin{cases} (w_{cycle} + 2w_{bdry})(\mathbf{A}^T\mathbf{y})_n = \left(1 - \tilde{\beta}\right)(\mathbf{A}^T\mathbf{y})_n & \text{if } n \le \frac{T}{2} \text{ or } n \ge N - \frac{T}{2} \\ (w_{bdry} + w_{cycle})2T - w_{cycle}(\mathbf{A}^T\mathbf{y})_n = \beta + \left(1 - 2\tilde{\beta}\right)(\mathbf{A}^T\mathbf{y})_n & \text{if } \frac{T}{2} \le n \le N - \frac{N}{2}. \end{cases} \tag{47}
$$

Therefore,

$$
-\frac{1}{\beta}\mathbf{A}^T(\mathbf{A}\mathbf{z} - \mathbf{y})_n = \begin{cases} \frac{1}{\beta_c}(\mathbf{A}^T\mathbf{y})_n & \text{if } n \le \frac{T}{2} \text{ or } n \ge N - \frac{T}{2} \\ 1 + \frac{2}{\beta_c}(\mathbf{A}^T\mathbf{y})_n & \text{if } \frac{T}{2} \le n \le N - \frac{N}{2}. \end{cases} \tag{48}
$$

By Lemma 28, $\beta_c = T$. By Lemma 26, $0 \le \frac{1}{\beta_c}(\mathbf{A}^T\mathbf{h}^{(T)})_n = \frac{1}{\beta_c}(\mathbf{A}^T\mathbf{y})_n \le 1$, and $\frac{1}{\beta_c}(\mathbf{A}^T\mathbf{y})_n = 1$ when n is an odd multiple of $\frac{T}{2}$. Since $0 < \beta < \frac{\beta_c}{2}$, we have $w_{bdry} > 0$ and $w_{cycle} < 0$. Therefore $\frac{1}{\beta}\mathbf{A}^T(\mathbf{A}\mathbf{z} - \mathbf{y})_n = -\text{sign}(\mathbf{z}^*_n)$ when $\mathbf{z}^*_n \neq 0$, ie n is an integer multiple of $\frac{T}{2}$. And for all $n \in [N]$, $\left|\frac{1}{\beta}\mathbf{A}^T(\mathbf{A}\mathbf{z} - \mathbf{y})_n\right| \le 1$. By Remark 19, $\mathbf{z}^*$ is optimal. ■

Figure 16 illustrates an example of $\frac{1}{2w_{bdry}}\mathbf{A}^T\mathbf{A}\mathbf{z_{bdry}}$ (44) and $\frac{1}{w_{cycle}}\mathbf{A}^T\mathbf{A}\mathbf{z_{cycle}}$.

Proof [Theorem 4] Summarizing Lemma 29 and Lemma 31 gives

$$
\mathbf{z}^* = \begin{cases} \frac{1}{2}\left(1 - \tilde{\beta}\right)_+\left(\mathbf{e}^{\left(\frac{T}{2}\right)} + \mathbf{e}^{\left(N - \frac{T}{2}\right)}\right) & \text{if } \tilde{\beta} \ge \frac{1}{2} \\ \mathbf{z_{bdry}} + \mathbf{z_{cycle}} & \text{if } 0 < \tilde{\beta} \le \frac{1}{2}. \end{cases}
$$

■

Proof [Corollary 3] Note that unscaling (defined in Section 2) scales the parameters in θ but does not change the neural network as a function. The reconstructed neural net (Definition 6) before unscaling is $f_2(x;\theta) = \sum_{i=1}^{N} z_i^* \sigma(x - x_i)$. For $\frac{1}{2} \leq \tilde{\beta} \leq 1$, $f_2(x;\theta) = \frac{1}{2}\left(1-\tilde{\beta}\right)_+ \left(\sigma\left(x - x_{\frac{T}{2}}\right) + \sigma\left(x - x_{N-\frac{T}{2}}\right)\right)$. We can compute $f_2(x;\theta)$ similarly for $\tilde{\beta} \leq \frac{1}{2}$. ■

G.2 Assume $L = 3$ layers

Proof [Theorem 5] Note $\mathbf{y} = \mathbf{h}^{(T)}$ switches $2k-1$ times. So by Theorem 2, there is an index i for which $\mathbf{A}_i = \mathbf{h}^{(T)}$. Since $\mathbf{y} = -\mathbf{A}_i$, and for all $n \in [N], \|\mathbf{A}_n\|_2 = N$, we have $i \in \text{argmax}_{n\in N} |\mathbf{A}_n^T \mathbf{y}|$. By Remark 22, $\beta_c = \max_{n\in N} |\mathbf{A}_n^T \mathbf{y}| = \mathbf{y}^T \mathbf{A}_i = N$. So when $\beta > \beta_c$, $\mathbf{z}^* = 0$, consistent with $z_i = -\left(1-\tilde{\beta}\right)_+$.

Next, $\mathbf{z}^*$ satisfies the subgradient condition in Remark 19, since for $n \in [N]$, $\left|\mathbf{A}_n^T(\mathbf{A}\mathbf{z}^* - \mathbf{y})\right| = \left|\mathbf{A}_n^T(z_i\mathbf{A}_i - \mathbf{A}_i)\right| = (z_i - 1)\left|\mathbf{A}_n^T \mathbf{A}_i\right| = \left|\frac{\beta}{\beta_c}\mathbf{A}_n^T \mathbf{y}\right| \leq \frac{\beta}{\beta_c}\text{argmax}_{n\in N}\left|\mathbf{A}_n^T \mathbf{y}\right| =\leq \beta$. Since $\mathbf{z}_i^* < 0$, when $i = n$, $\mathbf{A}_i^T(\mathbf{A}\mathbf{z}^* - \mathbf{y}) = \beta = -\beta\text{sign}(\mathbf{z}_i^*)$. By Remark 19, $\mathbf{z}^*$ is optimal. ■

Proof [Corollary 4] Follows from the reconstruction in Lemma 3. ■

Appendix H. The solution sets of Lasso and the training problem

Proof [Proposition 1] The result is almost a sub-case of that given by Mishkin and Pilanci (Mishkin and Pilanci, 2023) with the exception that the bias parameter ξ, is not regularized. Therefore optimality conditions do not impose a sign constraint and it is sufficient that $\mathbf{1}^\top(\mathbf{A}\mathbf{z} + \xi\mathbf{1} - \mathbf{y}) = 0$ for ξ to be optimal. This stationarity condition is guaranteed by $\mathbf{A}\mathbf{z} + \xi\mathbf{1} = \hat{\mathbf{y}}$. Now let us look at the parameters z_i. If $i \notin \mathcal{E}(\beta)$, then $z_i = 0$ is necessary and sufficient from standard results on the Lasso (Tibshirani, 2013). If $i \in \mathcal{E}(\beta)$ and $z_i \neq 0$, then $\mathbf{A}_i^\top(\hat{\mathbf{y}} - \mathbf{y}) = \beta\text{sign}(z_i)$, which shows that $\mathbf{z}_i$ satisfies first-order conditions. If $z_i = 0$, then first-order optimality is immediate since $|\mathbf{A}_i^\top(\hat{\mathbf{y}} - \mathbf{y})| \leq \beta$, holds. Putting these cases together completes the proof. ■

Proof [Proposition 2] This result follows from applying the reconstruction in Definition 6 to each optimal point in Φ. The reconstuction sets $\alpha_i = \text{sign}(z_i)\sqrt{|z_i|}$. From this we deduce $\text{sign}(\alpha_i) = \text{sign}\left(\mathbf{A}_i^\top(\mathbf{y} - \hat{\mathbf{y}})\right)$. The solution mapping determines the values of w_i and b_i and in terms of α_i. Finally, the constraint $f_2(\mathbf{X};\theta) = \hat{\mathbf{y}}$ follows immediately by equality of the convex and non-convex prediction functions on the training set. ■

Lemma 32 *Suppose $L = 2$, and the activation is ReLU, leaky ReLU or absolute value. Suppose $m^* \leq m \leq |\mathcal{M}|$. Since $m \leq |\mathcal{M}|$, we can map the set of neuron indices to $\mathcal{M}$. Let*

$\Theta^{Lasso,stat} = \{\theta : \forall i = (\mathbf{s}, \mathbf{j}, k) \in \mathcal{M}, b_i = -x_j w_i\} \subset \Theta$. *Let* $\theta^* \in \Theta^{Lasso,stat} \cap C(\beta)$. *Then* θ^* *is a minima of the Lasso problem.*

Proof Since $m^* \leq m$, the reconstructed neural net is optimal in the training problem. Let $F(\theta)$ and $F^{Lasso}(\mathbf{z}, \xi)$ be the objectives of the non-convex training problem (1) and Lasso (2), respectively. The parameters θ are stationary if $\theta \in C(\beta)$, i.e., $0 \in \partial F(\theta)$.

Let $\Theta^{Lasso} = \{\theta : \forall i = (\mathbf{s}, \mathbf{j}, k) \in \mathcal{M}, |w_i| = |\alpha_i|, b_i = -x_j w_i\} \subset \Theta^{Lasso,\text{stat}}$. By a similar argument as the proof of Theorem 3 in Wang et al. (2021), since $0 \in \partial F(\theta^*)$, we have $|w_i| = |\alpha_i|$ for all neurons i. Therefore $\theta^* \in \Theta^{Lasso}$. Then $\theta^* = (\boldsymbol{\alpha}^*, \xi^*, \mathbf{w}^*, \mathbf{b}^*) = R^{\alpha, \xi \to \theta}(\boldsymbol{\alpha}^*, \xi^*)$. Let $\tilde{F}(\boldsymbol{\alpha}, \xi) = F\left(R^{\alpha, \xi \to \theta}(\boldsymbol{\alpha}, \xi)\right)$.

Since $0 \in \partial F(\theta^*)$ at $(\boldsymbol{\alpha}^*, \xi^*) = \left(R^{\alpha, \xi \to \theta}\right)^{-1}(\theta^*)$, we have $\tilde{F}(\boldsymbol{\alpha}^*, \xi^*) = F(\theta^*)$. The chain rule gives $\partial_{(\boldsymbol{\alpha}, \xi)} \tilde{F}(\boldsymbol{\alpha}^*, \xi^*) = \partial_\theta F(\theta^*) \partial_{(\boldsymbol{\alpha}, \xi)} R^{\alpha, \xi \to \theta}(\boldsymbol{\alpha}^*, \xi^*) \ni \mathbf{0}$. Let $R^{\alpha \to z}(\boldsymbol{\alpha}) = \text{sign}(\boldsymbol{\alpha}) \boldsymbol{\alpha}^2$, with operations done elementwise. Next, at $(\mathbf{z}^*, \xi^*) = (R^{\alpha \to z}(\boldsymbol{\alpha}^*), \xi^*)$, we have $F^{Lasso}(\mathbf{z}^*, \xi^*) = \tilde{F}(\boldsymbol{\alpha}^*, \xi^*)$. The chain rule gives $\partial_{(\mathbf{z}, \xi)} F^{Lasso}(\mathbf{z}^*, \xi^*) = \partial_{(\boldsymbol{\alpha}, \xi)} \tilde{F}(\boldsymbol{\alpha}^*, \xi^*) \partial_{(\mathbf{z}, \xi)} R(\mathbf{z}^*, \xi^*) \ni \mathbf{0}$. Since the Lasso problem (2) is convex, the result holds. ■

Proof [Proposition 3] Observe that $R(\Phi(\beta)) \subseteq \tilde{\mathcal{C}}(\beta) \cap \Theta^{Lasso,stat} \subseteq \mathcal{C}(\beta) \cap \Theta^{Lasso,stat} \subseteq R(\Phi(\beta))$, where the first and last subset inequality follow from Theorem 1 and Lemma 32, respectively. Therefore all subsets in the above expression are equal. Observe that $P\left(\Theta^{Lasso,stat}\right) = \Theta^P$ and so $P\left(C(\beta) \cap \Theta^{Lasso,stat}\right) = C(\beta) \cap P\left(\Theta^{Lasso,stat}\right) = C(\beta) \cap \Theta^P$ and similarly $P\left(\tilde{C}(\beta) \cap \Theta^{Lasso,stat}\right) = C\tilde{(\beta)} \cap \Theta^P$. Now apply P to all subsets above. ■

Appendix I. Numerical results

I.1 Toy problems

In Figure 17 and Figure 18, in contrast to Figure 2, we *numerically* solve deep narrow ReLU Lasso problems (not the min-norm version) with $\beta = 10^{-8}$ for two other datasets and plot the reconstructed neural nets. These examples show that when the regularization is small enough, due to numerical solver tolerance, we may find near-optimal (but not optimal) neural nets, and even these near-optimal solutions exhibit the deep narrow features expected for each depth. Our simulations compute a near-optimal Lasso solution up to tolerance 10^{-8}.

In these Lasso simulations, $\beta \approx 0$, and the neural net fits the training data almost exactly, so the Lasso problem has approached the minimum norm regime (9). By Lemma 5, in Figure 17 and Figure 18, the $L = 3$ Lasso solution found numerically is optimal in both the $L = 3$ and $L = 4$ min norm problems, while the $L = 4$ Lasso solution found numerically is not, giving a higher objective $\|\mathbf{z}^*\|_1$. Moreover, in Figure 17, for $L = 2, 3$, any solution for depth L is also a solution for layer $L + 1$. In Figure 18, we also solve standard non-convex training problems with $m_L = 100$. The number of training epochs is $1,000$ for $L = 3$ and $50,000$ for $L = 4$. Adam was used with a learning rate of $0.5 \cdot 10^{-2}$.

As seen from Figure 17 and Figure 18, when $L \in \{2, 3\}$, the breakpoints in the Lasso and non-convex solutions occur at the training data points. When $L = 4$, the Lasso and non-convex solutions found numerically, even though they are not optimal, have reflection breakpoints as characteristic of the $L = 4$ dictionary, namely at $4 = R_{(0,2)}$.

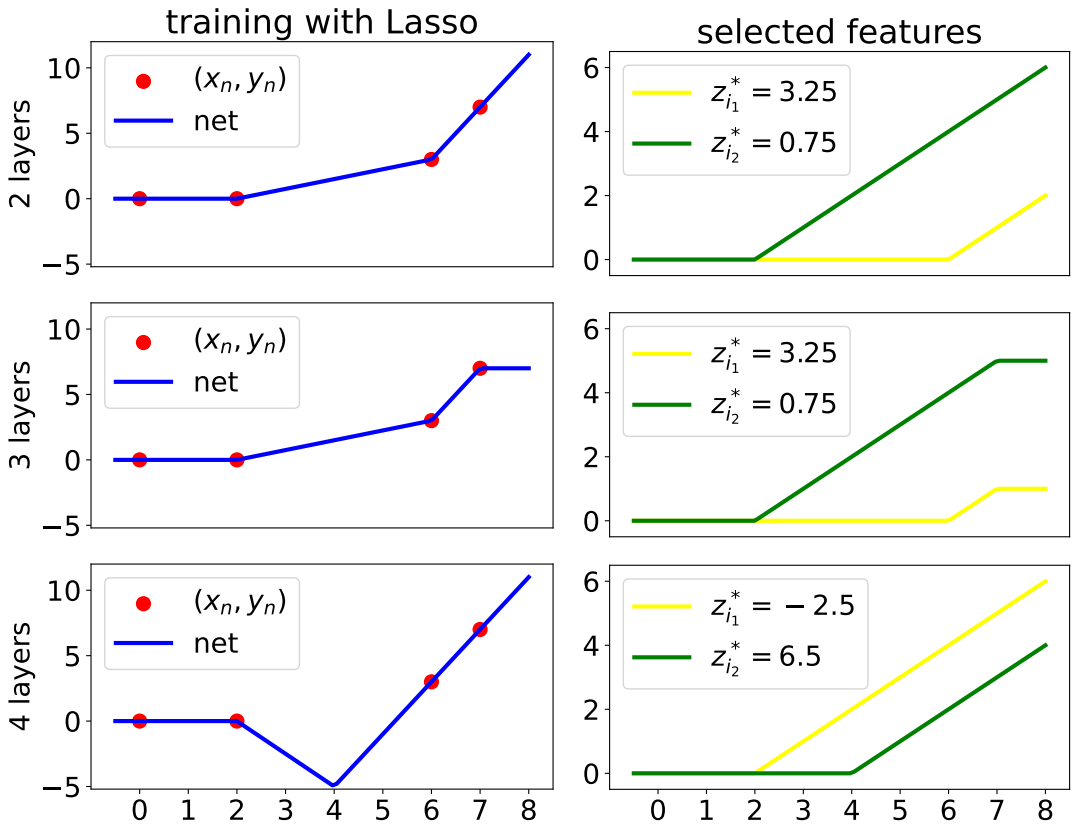


Figure 17: Left column: Predictions of deep narrow networks (blue) with $L = 2, 3$ and 4 layers trained using the Lasso problem with cvxpy on the same 1-D dataset $\mathbf{X} = (0, 2, 6, 7)^T, \mathbf{y} = (0, 0, 3, 7)^T$ (red dots). Right column: Lasso features for nonzero z_i^*, where $\mathbf{z}^*$ is the Lasso solution that is found.

I.2 Autoregression figures

In all figures except for the regularization path, the horizontal axis is the training epoch.

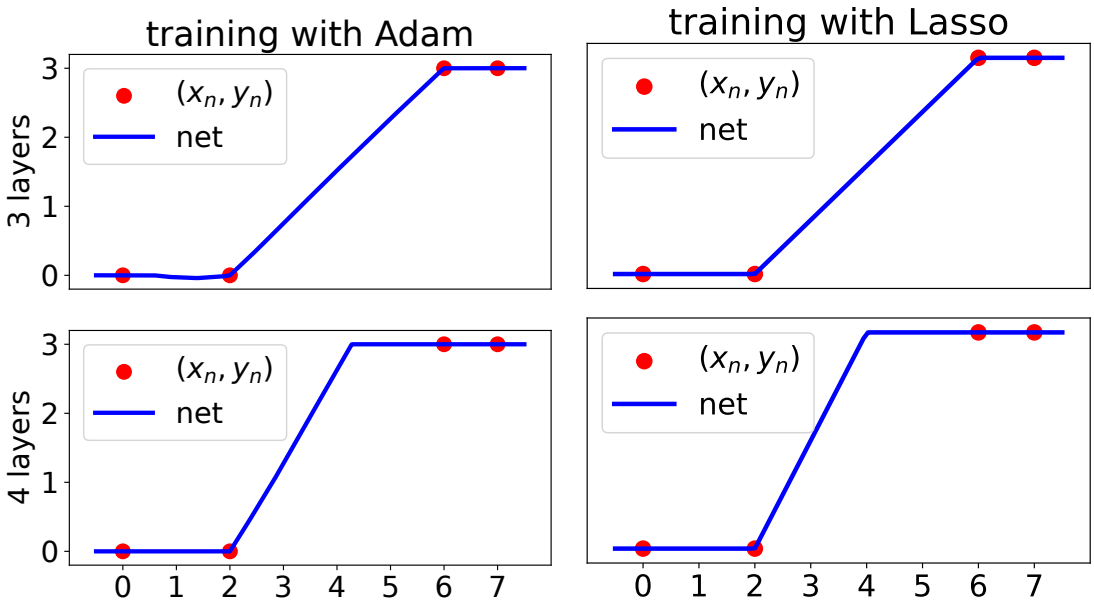
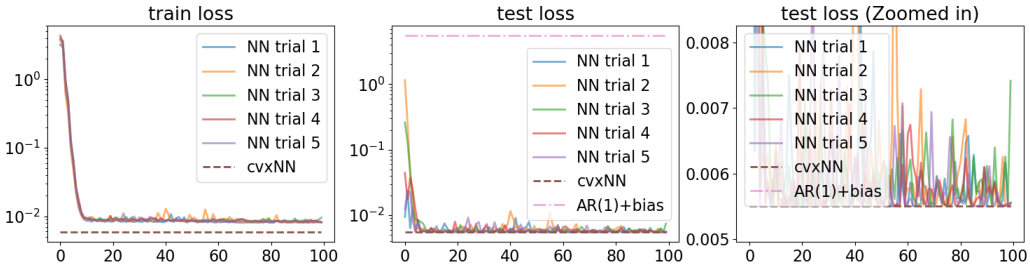
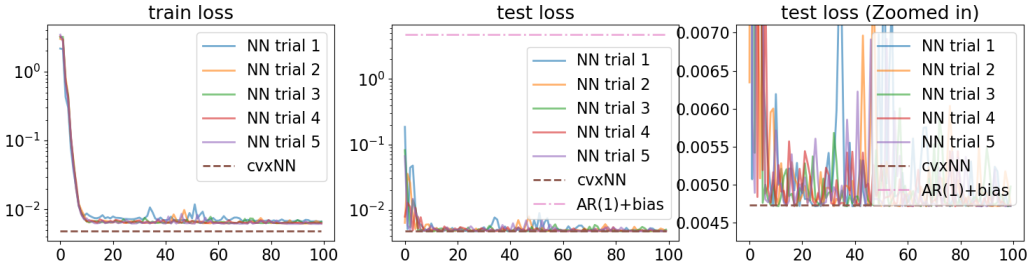


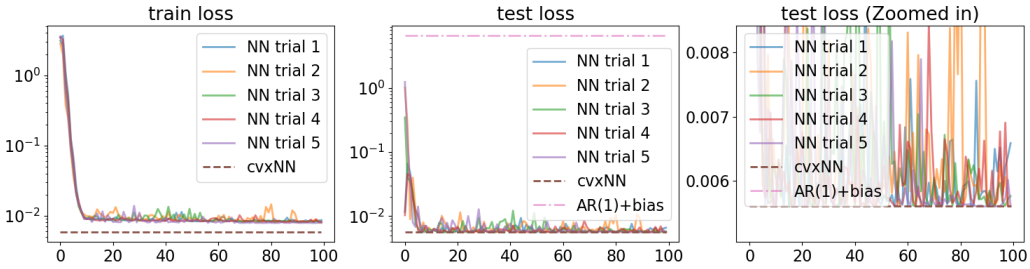
Figure 18: Predictions of parallel networks (blue) with $L = 3$ and 4 layers are trained on the same 1-D dataset $\mathbf{X} = (0, 2, 6, 7)^T, \mathbf{y} = (0, 0, 3, 3)^T$ (red dots) using the non-convex training problem (left) and Lasso problem (right).



$m = 2$



$m = 5.$



$m = 10.$

Figure 19: Planted data. $\sigma^2 = 1$.

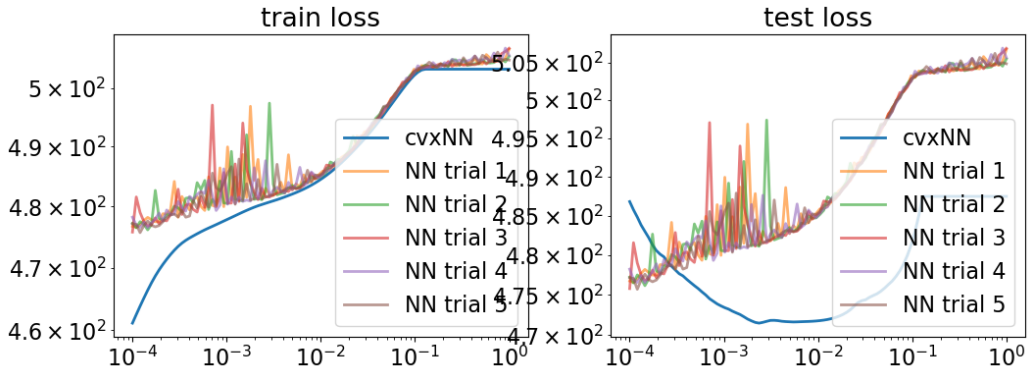
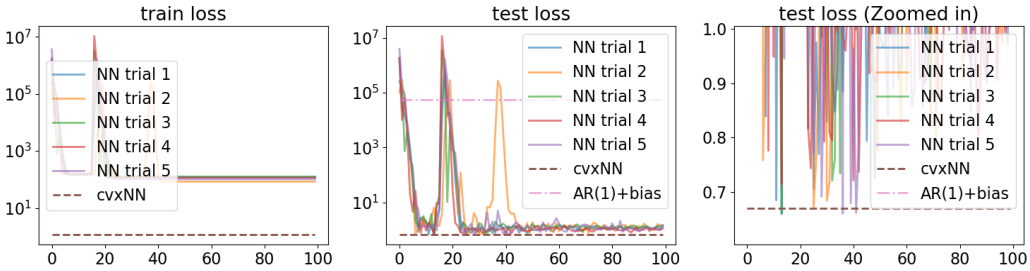
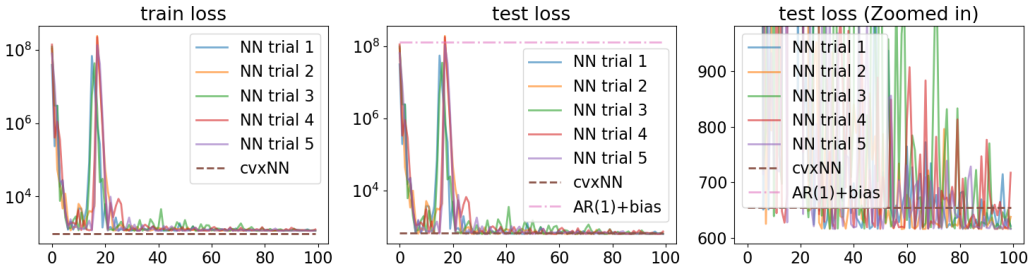


Figure 20: The regularization path. Here, $\sigma^2 = 1$, $m = 5$.

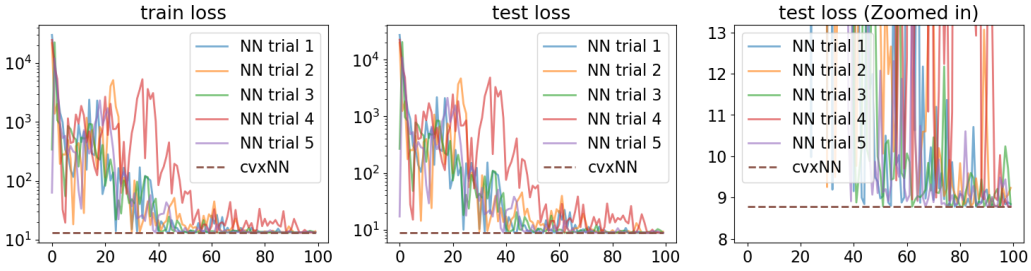


BTC-2017min.



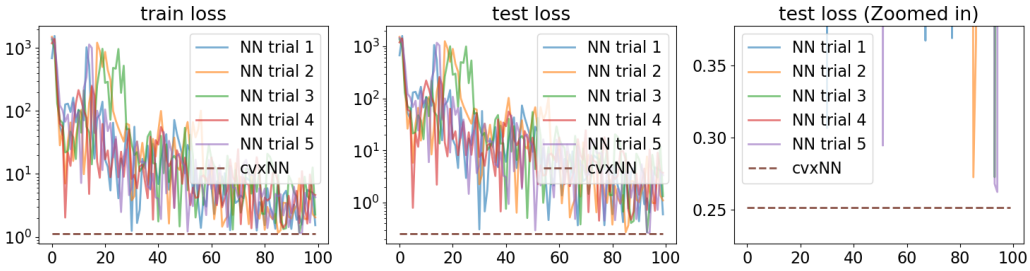
BTC-hourly.

Figure 21: Regression with L2 loss.



BTC-hourly.

Figure 22: Regression with quantile loss. $\tau = 0.3$



BTC-2017min.

Figure 23: Regression with quantile loss. $\tau = 0.7$