Convex Geometry and Duality of Over-parameterized Neural Networks

Tolga Ergen and Mert Pilanci
Department of Electrical Engineering
Stanford University
Stanford, CA 94305, USA
{ergen, pilanci}@stanford.edu

Abstract

We develop a convex analytic framework for ReLU neural networks which elucidates the inner workings of hidden neurons and their function space characteristics. We show that neural networks with rectified linear units act as convex regularizers, where simple solutions are encouraged via extreme points of a certain convex set. For one dimensional regression and classification, as well as rank-one data matrices, we prove that finite two-layer ReLU networks with norm regularization yield linear spline interpolation. We characterize the classification decision regions in terms of a closed form kernel matrix and minimum L1 norm solutions. This is in contrast to Neural Tangent Kernel which is unable to explain neural network predictions with finitely many neurons. Our convex geometric description also provides intuitive explanations of hidden neurons as auto-encoders. In higher dimensions, we show that the training problem for two-layer networks can be cast as a finite dimensional convex optimization problem with infinitely many constraints. We then provide a family of convex relaxations to approximate the solution, and a cutting-plane algorithm to improve the relaxations. We derive conditions for the exactness of the relaxations and provide simple closed form formulas for the optimal neural network weights in certain cases. We also establish a connection to ℓ_0-ℓ_1 equivalence for neural networks analogous to the minimal cardinality solutions in compressed sensing. Extensive experimental results show that the proposed approach yields interpretable and accurate models.

1 Introduction

Understanding the fundamental reason why training over-parameterized Deep Neural Networks (DNNs) converges to minimizers that generalize well remains an open problem. Recently, it was empirically observed that ReLU NNs exhibit an interesting structure, where only finitely many simple functions can be obtained as optimal solutions [1, 2]. In [2], the function space of one dimensional (1D) ReLU regression networks was studied, where it was shown that among infinitely many two-layer ReLU networks that perfectly fit the training data, the one with the minimum Euclidean norm parameters yields a linear spline interpolation. It is possible that the structure induced by over-parameterization explains remarkable generalization properties of DNNs. Despite the dramatic surge of interest in NNs, the fundamental mechanism behind these simple structures in over-parameterized networks is largely unknown.

In this paper, we develop a convex analytic framework to analyze two-layer ReLU NNs and characterize the structure that emerges as a result of over-parameterization. We show that over-parameterized networks behave like *convex regularizers*, where simple solutions are encouraged via *extreme points* of a certain convex set. Our results are analogous to ℓ_1-norm regularization

where sparse solutions are encouraged as a result of the 1-sparse extreme points of the ℓ_1-ball. However, unlike these methods, we show that the extreme points in over-parameterized NNs are data-adaptive and precisely serve as *convex autoencoders*. We fully describe the extreme points via analytical expressions. In one dimensional regression and classification, the extreme points manifest as finitely many simple quantized solutions, and yield linear spline interpolations for regression/classification tasks explaining and extending recent results.

1.1 Related work

A line of research [1, 3, 4] explored the behavior of ReLU networks in finite size cases. In [4], the authors indicated that NNs are implicitly regularized during training since Stochastic Gradient Descent (SGD) converges to a solution with small norm. The idea of implicit regularization was also extended to the networks trained with GD as well as SGD. Particularly, the authors in [1] showed that implicit regularization has a strong connection with the initialization of a network and proved that network weights tend to align along certain directions determined by the input data, which implies that there are only finitely many possible simple functions for a given dataset. In order to explain generalization capabilities of ReLU networks, another line of research in [5, 6, 7, 8] focused on infinitely wide two-layer ReLU networks. In [5], the authors introduced an algorithm that can train a regularized NN with infinite width in an incremental manner. In [6], the authors adopted a margin-based perspective, where they showed that the optimal point of a weakly regularized loss has the maximum margin property, thus, over-parameterization can improve generalization bounds. Furthermore, recently, the connection between infinite width NNs and kernel methods has also attracted significant attention [9, 10]. Such kernel based methods, nowadays known as Neural Tangent Kernel (NTK), work in a regime where the parameters barely move, i.e., the lazy regime, so that the dynamics of an NN training problem can be characterized via a deterministic fixed kernel matrix. Therefore, these studies showed that an NN trained with GD and infinitesimal step size in the lazy regime is equivalent to a kernel predictor with a fixed kernel matrix.

1.2 Our contributions

Our contributions can be summarized as follows.

- We develop a convex analytic framework for two-layer ReLU NNs to provide a deeper insight into over-parameterization and implicit regularization. We show that over-parameterized NNs behave like convex regularizers, where simple structures are encouraged in the solution via the extreme points of a well-defined regularizer.

- For one dimensional regression and classification, we prove that hidden layers form a linear spline interpolation. We also provide an intuitive convex geometric explanation of this fact, and derive a general formula for the hidden layer representation in higher dimensions.

- Using our convex analytic framework, we characterize a finite set of optimal solutions in some specific cases so that the training problem can be reformulated as a convex optimization problem. We also prove that there might exist multiple optimal solutions in these cases.

- We provide a convex relaxation based training procedure, which is proven to be exact under certain assumptions on the training set. We also prove that these assumptions hold in generic regimes, e.g., when the training examples are i.i.d. random.
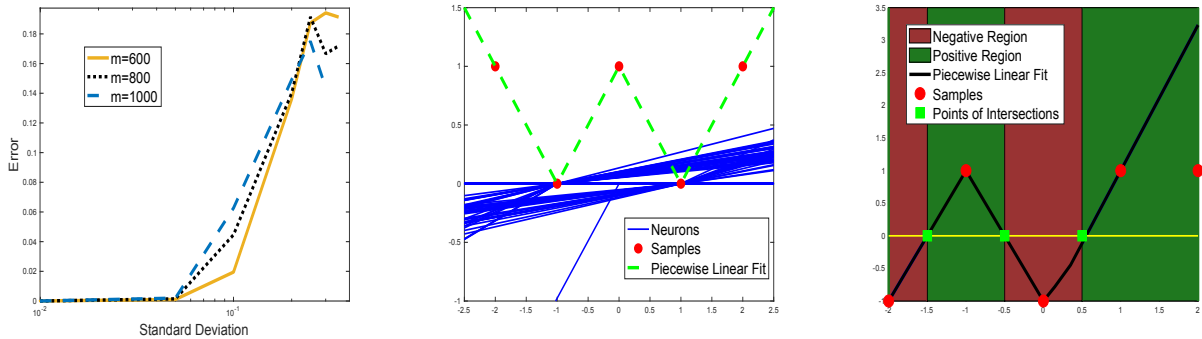
(a) Deviation of the ReLU network output from piecewise linear spline vs standard deviation of initialization plotted for different number of hidden neurons m.

(b) Contribution of each neuron along with the overall fit. Each activation point corresponds to a particular data sample.

(c) Binary classification using hinge loss. Network output is a linear spline interpolation, and decision regions are determined by zero crossings (see Lemma 2.6).

Figure 1: Analysis of one dimensional regression and classification with a two-layer NN.

- We establish an ℓ_0-ℓ_1 equivalence for NNs, which parallels minimum cardinality relaxations in compressed sensing. We then provide closed form expressions for the optimal ReLU networks in certain cases.

1.3 Overview of our results

Implicit regularization plays a key role in training NNs, however, it is still theoretically elusive how NNs trained with gradient descent (GD) and no regularization obtain simple solutions, e.g., spline interpolation in 1D datasets. In order to gain a deeper insight into the effects of initialization magnitude, we perform a simple experiment on training 1D ReLU NNs on the data shown in Figure 1b. The results in Figures 1a and 1b show that the two-layer ReLU regression network fits precisely a linear spline interpolation when the standard deviation of the (zero-mean) initialization is below a critical value. Thus, as emphasized by [1, 8], initialization magnitude is critical for the final norm of the network parameters, so that GD converges solutions with smaller norm, i.e., closer to initialization, which can generalize as a result of this implicit regularization. In Figure 1b, we also display the set of neurons found by GD and the corresponding overall function fit in the case of small initialization. In this over-parameterized scenario, linear combination of the neurons with different weights and biases still outputs a linear spline interpolation. The same results also hold for the binary classification using a two-layer ReLU network with the hinge loss as illustrated in Figure 1c. The network fits a certain piecewise linear function and the decision region (to label the samples as ± 1) boundaries become precisely the zero crossings of this function. The central questions we will address in this paper are: *Why are over-parameterized NNs providing a linear spline interpolation in 1D? Is there a general mechanism encouraging simple solutions in arbitrary dimensions? How are the decision regions formed ?* We show that these questions can be rigorously answered using *convex geometry* and *duality*.

Simply stated, we show that the optimal solutions have kinks at input data points precisely

because the convex approximation[1] of a data point x_i given by

$$\min_{\boldsymbol{\lambda} \succcurlyeq 0, \sum_j \lambda_j = 1} \left| x_i - \sum_{j \in \mathcal{S}, j \neq i} \lambda_j x_j \right|$$

is given by another data point, i.e., an extreme point of the convex hull of data points in $\mathcal{S} \backslash \{i\}$. Consequently, input data points are the only allowed hidden neuron activation thresholds at optimum for 1D regression and classification networks. We further provide a general formula for the hidden neuron configurations in the multivariate case.

Specifically, we focus on minimizers for two-layer NNs with small Euclidean norm. In one dimensional data sets, through convex analytic arguments, we establish that each training sample becomes an extreme point of a certain convex set, which means that the activation threshold of a ReLU function has to correspond to one of the data samples. This result completely explains what we observe in Figure 1. Since the data samples are the activation thresholds, we observe a piecewise linear function as the NN output, where the kinks occur exactly at the data samples (activation points). Our analysis also reveals that the hidden neurons can be interpreted as data autoencoders in higher dimensions and they can further be expressed in closed form in certain cases.

We also provide a *representer theorem* for the optimal neurons in a general two-layer NN. In particular, in a finite training dataset with samples $\mathbf{a}_1, \ldots, \mathbf{a}_n \in \mathbb{R}^d$, the hidden neurons $\mathbf{u}_1, \ldots, \mathbf{u}_m \in \mathbb{R}^d$ obey

$$\mathbf{u}_j = \sum_i \alpha_i (\mathbf{a}_i - \mathbf{a}_k) \text{ and } b = -\mathbf{a}_k^T \mathbf{u}_j \ \forall j \in [m],$$

for some weight vector $\boldsymbol{\alpha}$ and index $k \in [m]$.

Notation: We denote the matrices and vectors as uppercase and lowercase bold letters, respectively. To denote a vector or matrix of zeros or ones, we use $\mathbf{0}$ or $\mathbf{1}$, respectively, where the sizes are understood from the context. Additionally, $\mathbf{I}_k$ represents the identity matrix of the size k. We also use $(x)_+ = \max\{x, 0\}$ for the ReLU activation. Furthermore, we denote the set of integers from 1 to n as $[n]$ and use the notation $\mathbf{e}_1, \ldots \mathbf{e}_n$ for the ordinary basis vectors in $\mathbb{R}^n$.

2 Preliminaries

Given n data samples, i.e., $\{\mathbf{a}_i\}_{i=1}^n, \mathbf{a}_i \in \mathbb{R}^d$, we consider two-layer NNs with m hidden neurons and ReLU activations. Initially, we focus on the scalar output case for simplicity, i.e., [2]

$$f(\mathbf{A}) = \sum_{j=1}^m w_j (\mathbf{A}\mathbf{u}_j + b_j \mathbf{1})_+, \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{n \times d}$ is the data matrix, $\mathbf{u}_j \in \mathbb{R}^d$ and $b_j \in \mathbb{R}$ are the parameters of the jth hidden neuron, and w_j's are the weights for the output layer. For a more compact representation, we also define $\mathbf{U} \in \mathbb{R}^{d \times m}$, $\mathbf{b} \in \mathbb{R}^m$, and $\mathbf{w} \in \mathbb{R}^m$ as the hidden layer weight matrix, the bias vector, and the output layer weight vector, respectively. Thus, (1) can be written as $f(\mathbf{A}) = (\mathbf{A}\mathbf{U} + \mathbf{1}\mathbf{b}^T)_+ \mathbf{w}$.[3]

---

[1]Here x_i is an arbitrary data sample, and $\mathcal{S}$ is an arbitrary subset of data points. $\lambda_1, \ldots, \lambda_n$ are mixture weights, approximating x_i as a convex mixture of the data points in $\mathcal{S} \backslash \{i\}$.

[2]We assume that the bias term for the output layer is zero without loss of generality, since we can still recover the general case as illustrated in [1].

[3]We emphasize that all the derivations in the sequel can be extended to a vector case with o outputs. We refer the reader to the supplementary material for further details.

Given the data matrix $\mathbf{A}$ and the label vector $\mathbf{y} \in \mathbb{R}^n$, consider training the network by solving the following optimization problem

$$\min_{w_j, \mathbf{u}_j, b_j} \left\| \sum_{j=1}^{m} w_j (\mathbf{A}\mathbf{u}_j + b_j \mathbf{1})_+ - \mathbf{y} \right\|_2^2 + \beta \sum_{j=1}^{m} (\|\mathbf{u}_j\|_2^2 + w_j^2), \tag{2}$$

where β is a regularization parameter. We define the overall parameter space Θ for (1) as $\theta \in \Theta = \{(\mathbf{U}, \mathbf{b}, \mathbf{w}, m) \mid \mathbf{U} \in \mathbb{R}^{d \times m}, \mathbf{b} \in \mathbb{R}^m, \mathbf{w} \in \mathbb{R}^m, m \in \mathbb{Z}_+\}$. Based on our observations in Figure 1a and the results in [2, 8, 11, 12], we first focus on a minimum norm[4] variant of (2). We define the squared Euclidean norm of the weights (without biases) as $R(\theta) = \|\mathbf{w}\|_2^2 + \|\mathbf{U}\|_F^2$. Then we consider the following optimization problem

$$\min_{\theta \in \Theta} R(\theta) \text{ s.t. } f_\theta(\mathbf{A}) = \mathbf{y}, \tag{3}$$

where the over-parameterization allows us to reach zero training error over $\mathbf{A}$ via the ReLU network in (1). The next lemma shows that the minimum squared Euclidean norm is equivalent to minimum ℓ_1 norm after a suitable rescaling. This result was also presented in [2, 11].

Lemma 2.1 ([2, 11]). *The following two optimization problems are equivalent:*

$$P^* = \min_{\theta \in \Theta} R(\theta) \text{ s.t. } f_\theta(\mathbf{A}) = \mathbf{y} \qquad = \qquad \min_{\theta \in \Theta} \|\mathbf{w}\|_1 \text{ s.t. } f_\theta(\mathbf{A}) = \mathbf{y}, \|\mathbf{u}_j\|_2 = 1, \forall j.$$

Lemma 2.2. *Replacing $\|\mathbf{u}_j\|_2 = 1$ with $\|\mathbf{u}_j\|_2 \leq 1$ does not change the value of the above problem.*

By Lemma 2.1 and 2.2, we can express (3) as

$$\min_{\theta \in \Theta} \|\mathbf{w}\|_1 \text{ s.t. } f_\theta(\mathbf{A}) = \mathbf{y}, \|\mathbf{u}_j\|_2 \leq 1, \forall j. \tag{4}$$

However, both (2) and (4) are quite challenging optimization problems due to the complicated behavior of an affine mapping along with the ReLU activation. In particular, depending on the properties of $\mathbf{A}$, e.g., singular values, rank, and dimensions, the geometry of the objective in (2) might considerably change.

2.1 Geometry of a single ReLU neuron in the function space

In order to illustrate the geometry of (2), we particularly focus on a simple case where we have a single neuron with no bias and regularization, i.e., $m = 1$, $b_1 = 0$, and $\beta = 0$. Thus, (2) reduces to

$$\min_{\mathbf{u}_1} \left\| w_1 (\mathbf{A}\mathbf{u}_1)_+ - \mathbf{y} \right\|_2^2 \text{ s.t. } \|\mathbf{u}_1\|_2 \leq 1. \tag{5}$$

The solution of (5) is completely determined by the set $\mathcal{Q}_\mathbf{A} = \{(\mathbf{A}\mathbf{u})_+ \mid \mathbf{u} \in \mathbb{R}^d, \|\mathbf{u}\|_2 \leq 1\}$. It is evident that (5) is solved via scaling this set by $|w_1|$ to minimize the distance to $+\mathbf{y}$ or $-\mathbf{y}$, depending on the sign of w_1. We note that since $\|\mathbf{u}\|_2 \leq 1$ describes a d-dimensional unit ball, $\mathbf{A}\mathbf{u}$ describes an ellipsoid whose shape and orientation is determined by the singular values and the output singular vectors of $\mathbf{A}$ as illustrated in Figure 2.

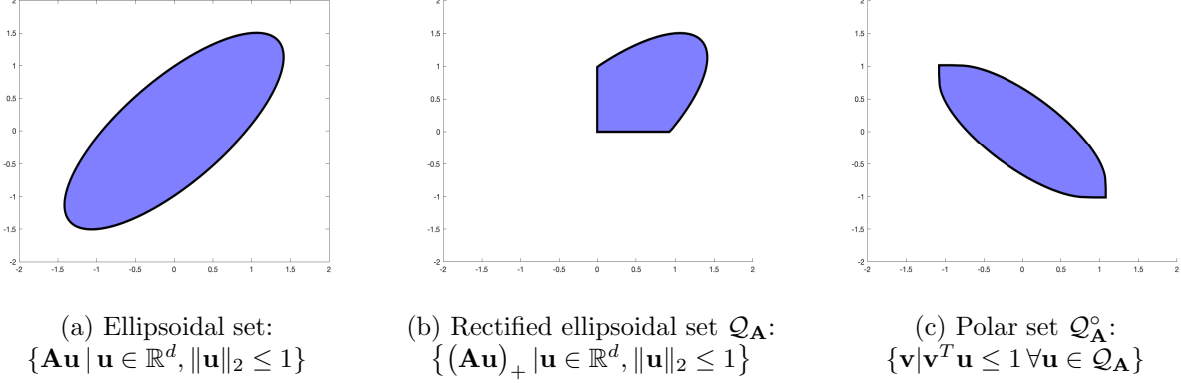[4]This corresponds to weak regularization, i.e., $\beta \to 0$ in (2) (see e.g. [6].).

(a) Ellipsoidal set:
$\{\mathbf{A}\mathbf{u} \,|\, \mathbf{u} \in \mathbb{R}^d, \|\mathbf{u}\|_2 \leq 1\}$

(b) Rectified ellipsoidal set $\mathcal{Q}_{\mathbf{A}}$:
$\left\{\left(\mathbf{A}\mathbf{u}\right)_+ \,|\, \mathbf{u} \in \mathbb{R}^d, \|\mathbf{u}\|_2 \leq 1\right\}$

(c) Polar set $\mathcal{Q}_{\mathbf{A}}^\circ$:
$\{\mathbf{v} \,|\, \mathbf{v}^T\mathbf{u} \leq 1 \,\forall \mathbf{u} \in \mathcal{Q}_{\mathbf{A}}\}$

Figure 2: Two dimensional illustration of a spike-free case. The extreme points (spikes) give rise to the linear spline interpolation behavior in Figures 1b and 1c as predicted by our theory (see Lemma 2.7). The set shown in the middle figure acts as a regularizer analogous to a non-convex atomic norm.

2.2 Rectified ellipsoid and its geometric properties

A central object in our analysis is the rectified ellipsoidal set introduced in the previous section, which is defined as $\mathcal{Q}_{\mathbf{A}} = \left\{\left(\mathbf{A}\mathbf{u}\right)_+ \,|\, \mathbf{u} \in \mathbb{R}^d, \|\mathbf{u}\|_2 \leq 1\right\}$. The set $\mathcal{Q}_{\mathbf{A}}$ is non-convex in general, as depicted in Figure 3, 4, and 5. However, there exists a family of data matrices $\mathbf{A}$ for which the set $\mathcal{Q}_{\mathbf{A}}$ is convex as illustrated in Figure 2, e.g., diagonal data matrices. However, the aforementioned set of matrices are, in fact, a more general class.

2.2.1 Spike-free matrices

We say that a matrix $\mathbf{A}$ is spike-free if it holds that $\mathcal{Q}_{\mathbf{A}} = \mathbf{A}\mathcal{B}_2 \cap \mathbb{R}_+^n$, where $\mathbf{A}\mathcal{B}_2 = \{\mathbf{A}\mathbf{u} \,|\, \mathbf{u} \in \mathcal{B}_2\}$, and $\mathcal{B}_2$ is the unit ℓ_2 ball defined as $\mathcal{B}_2 = \{\mathbf{u} \,|\, \|\mathbf{u}\|_2 \leq 1\}$. Note that $\mathcal{Q}_{\mathbf{A}}$ is a convex set if $\mathbf{A}$ is spike-free. In this case we have an efficient description of this set given by $\mathcal{Q}_{\mathbf{A}} = \{\mathbf{A}\mathbf{u} \,|\, \mathbf{u} \in \mathbb{R}^d, \|\mathbf{u}\|_2 \leq 1, \mathbf{A}\mathbf{u} \geq 0\}$.

If $\mathcal{Q}_{\mathbf{A}} = \{(\mathbf{A}\mathbf{u})_+ \,|\, \mathbf{u} \in \mathbb{R}^d, \|\mathbf{u}\|_2 \leq 1\}$ can be expressed as $\mathbb{R}_+^n \cap \{\mathbf{A}\mathbf{u} \,|\, \mathbf{u} \in \mathbb{R}^d, \|\mathbf{u}\|_2 \leq 1\}$ (see Figure 2), then (5) can be solved via convex optimization after the rescaling $\mathbf{u} = \mathbf{u}_1 w_1$

$$\min_{\mathbf{u}} \left\|\mathbf{A}\mathbf{u} - \mathbf{y}\right\|_2^2 \text{ s.t. } \mathbf{u} \in \{\mathbf{A}\mathbf{u} \succcurlyeq 0\} \cup \{-\mathbf{A}\mathbf{u} \succcurlyeq 0\}, \|\mathbf{u}\|_2 \leq 1\,.$$

The following lemma provides a characterization of spike-free matrices

Lemma 2.3. *A matrix $\mathbf{A}$ is spike-free if and only if the following condition holds*

$$\forall \mathbf{u} \in \mathcal{B}_2, \ \exists \mathbf{z} \in \mathcal{B}_2 \text{ such that we have } \left(\mathbf{A}\mathbf{u}\right)_+ = \mathbf{A}\mathbf{z}\,. \tag{6}$$

Alternatively, a matrix $\mathbf{A}$ is spike free if and only if it holds that

$$\max_{\mathbf{u}\,:\,\|\mathbf{u}\|_2 \leq 1, \,(\mathbf{I}_n - \mathbf{A}\mathbf{A}^\dagger)(\mathbf{A}\mathbf{u})_+ = \mathbf{0}} \|\mathbf{A}^\dagger\left(\mathbf{A}\mathbf{u}\right)_+\|_2 \leq 1\,.$$

If $\mathbf{A}$ is full row rank, then the above condition simplifies to

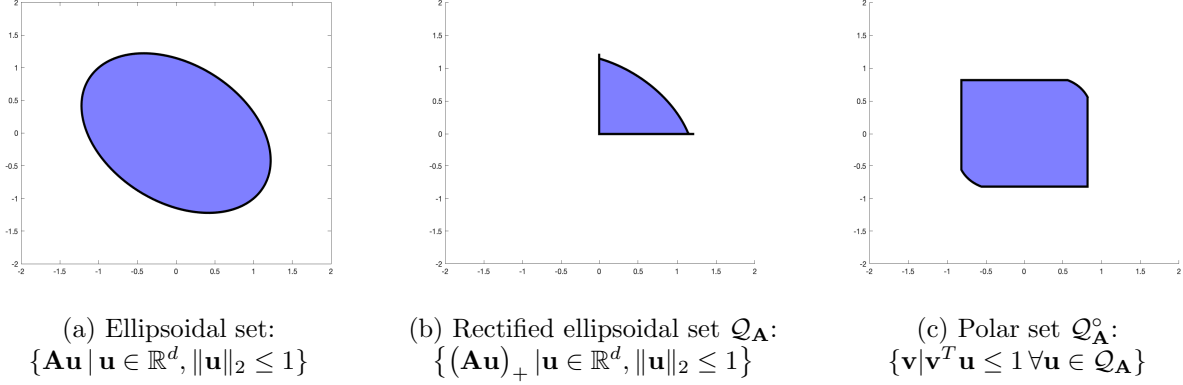$$\max_{\mathbf{u}\,:\,\|\mathbf{u}\|_2 \leq 1} \|\mathbf{A}^\dagger\left(\mathbf{A}\mathbf{u}\right)_+\|_2 \leq 1\,. \tag{7}$$

6

(a) Ellipsoidal set:
$\{\mathbf{A}\mathbf{u} \,|\, \mathbf{u} \in \mathbb{R}^d, \|\mathbf{u}\|_2 \le 1\}$

(b) Rectified ellipsoidal set $\mathcal{Q}_{\mathbf{A}}$:
$\left\{ (\mathbf{A}\mathbf{u})_+ \,|\, \mathbf{u} \in \mathbb{R}^d, \|\mathbf{u}\|_2 \le 1 \right\}$

(c) Polar set $\mathcal{Q}_{\mathbf{A}}^\circ$:
$\{\mathbf{v} \,|\, \mathbf{v}^T\mathbf{u} \le 1 \,\forall \mathbf{u} \in \mathcal{Q}_{\mathbf{A}}\}$

Figure 3: Two dimensional illustration of a the rectified ellipsoid that is not spike-free and its polar set. Note that the polar set exhibits a combination of smooth and non-smooth faces.

(a) Ellipsoidal set:
$\{\mathbf{A}\mathbf{u} \,|\, \mathbf{u} \in \mathbb{R}^d, \|\mathbf{u}\|_2 \le 1\}$

(b) Rectified ellipsoidal set $\mathcal{Q}_{\mathbf{A}}$:
$\left\{ (\mathbf{A}\mathbf{u})_+ \,|\, \mathbf{u} \in \mathbb{R}^d, \|\mathbf{u}\|_2 \le 1 \right\}$

(c) Polar set $\mathcal{Q}_{\mathbf{A}}^\circ$:
$\{\mathbf{v} \,|\, \mathbf{v}^T\mathbf{u} \le 1 \,\forall \mathbf{u} \in \mathcal{Q}_{\mathbf{A}}\}$
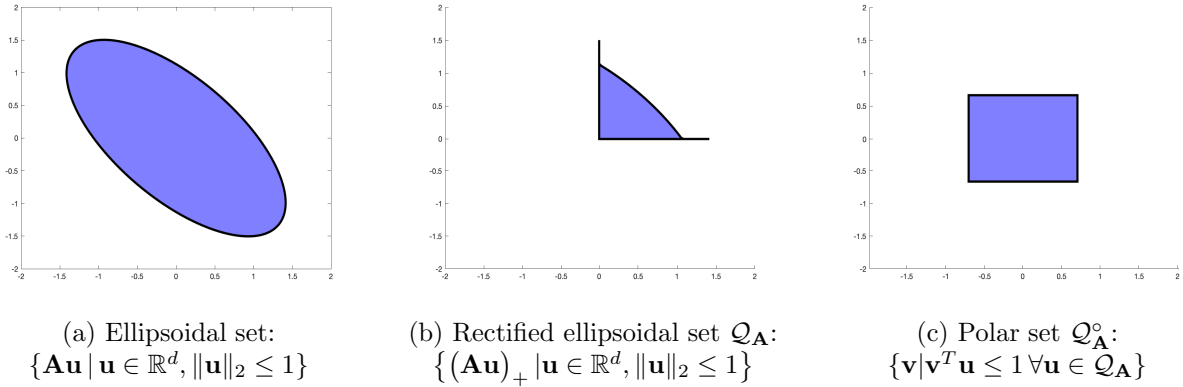
Figure 4: Two dimensional illustration of a the rectified ellipsoid that is not spike-free and its polar set. Note that the polar set is a rectangle since the convex hull of the rectified ellipsoid is a triangle.

We note that the condition in (7) bears a close resemblance to the irrepresentability conditions in Lasso support recovery (see e.g. [13]). It is easy to see that diagonal matrices are spike-free. More generally, any matrix of the form $\mathbf{A} = \mathbf{\Sigma}\mathbf{V}^T$, where $\mathbf{\Sigma}$ is diagonal, and $\mathbf{V}^T$ is any matrix with orthogonal rows, i.e., $\mathbf{V}^T\mathbf{V} = \mathbf{I}_n$, is spike-free. In other cases, $\mathcal{Q}_{\mathbf{A}}$ has a non-convex shape as illustrated in Figure 3 and 4. Therefore, the ReLU activation might exhibit significantly complicated and non-convex behavior as the dimensionality of the problem increases. Note that $\mathbf{A}\mathcal{B}_2 \cap \mathbb{R}_+^n \subseteq \mathcal{Q}_{\mathbf{A}}$ always holds, and therefore the former set is a convex relaxation of the set $\mathcal{Q}_{\mathbf{A}}$. We call this set spike-free relaxation of $\mathcal{Q}_{\mathbf{A}}$.

As another example for spike-free data matrices, we consider the Singular Value Decomposition of the data matrix $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ in compact form. We can apply a whitening transformation on the data matrix by defining $\tilde{\mathbf{A}} = \mathbf{A}\mathbf{V}\mathbf{\Sigma}^{-1}$, which is known as zero-phase whitening in the literature. Note that the empirical covariance of the whitened data is diagonal since we have $\tilde{\mathbf{A}}^T\tilde{\mathbf{A}} = \mathbf{I}_n$. Furthermore, rank-one data matrices with positive left singular vectors are also spike-free. The following lemmas prove these claims.

Lemma 2.4. *Let $\mathbf{A}$ be a whitened data matrix with $n \le d$ that satisfies $\mathbf{A}\mathbf{A}^T = \mathbf{I}_n$. Then, it holds*
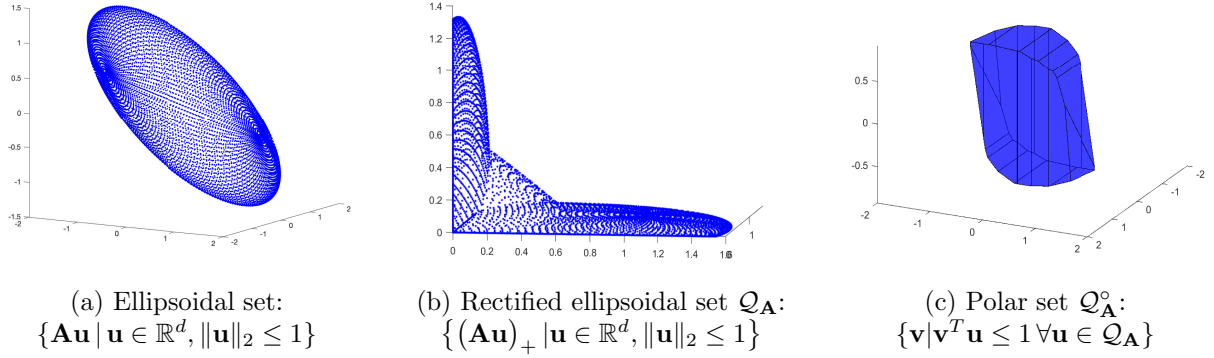
(a) Ellipsoidal set:
$\{\mathbf{A}\mathbf{u} \,|\, \mathbf{u} \in \mathbb{R}^d, \|\mathbf{u}\|_2 \leq 1\}$

(b) Rectified ellipsoidal set $\mathcal{Q}_{\mathbf{A}}$:
$\left\{(\mathbf{A}\mathbf{u})_+ \,|\, \mathbf{u} \in \mathbb{R}^d, \|\mathbf{u}\|_2 \leq 1\right\}$

(c) Polar set $\mathcal{Q}_{\mathbf{A}}^\circ$:
$\{\mathbf{v} \,|\, \mathbf{v}^T\mathbf{u} \leq 1 \,\forall \mathbf{u} \in \mathcal{Q}_{\mathbf{A}}\}$

Figure 5: Three dimensional illustration of the rectified ellipsoid and its polar set. Note that the polar set exhibits a combination of smooth and non-smooth faces.

that

$$\max_{\mathbf{u}\,:\,\|\mathbf{u}\|_2 \leq 1} \|\mathbf{A}^\dagger (\mathbf{A}\mathbf{u})_+\|_2 \leq 1\,,$$

where $\mathbf{A}^\dagger = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}$. As a direct consequence, $\mathbf{A}$ is spike-free.

Lemma 2.5. *Let $\mathbf{A}$ be a data matrix such that $\mathbf{A} = \mathbf{c}\mathbf{a}^T$, where $\mathbf{c} \in \mathbb{R}_+^n$ and $\mathbf{a} \in \mathbb{R}^d$. Then, $\mathbf{A}$ is spike-free.*

2.3 Polar convex duality

It can be shown that the dual of the problem (4) is given by[5]

$$\max_{\mathbf{v}} \mathbf{v}^T\mathbf{y} \text{ s.t. } \mathbf{v} \in \mathcal{Q}_{\mathbf{A}}^\circ\,, \ -\mathbf{v} \in \mathcal{Q}_{\mathbf{A}}^\circ, \tag{8}$$

where $\mathcal{Q}_{\mathbf{A}}^\circ$ is the polar set [14] of $\mathcal{Q}_{\mathbf{A}}$ defined as $\mathcal{Q}_{\mathbf{A}}^\circ = \{\mathbf{v} \,|\, \mathbf{v}^T\mathbf{u} \leq 1, \,\forall \mathbf{u} \in \mathcal{Q}_{\mathbf{A}}\}\,.$

2.4 Extreme points

Extreme points of a convex set $\mathcal{C}$ is defined as the set of points $\mathbf{v} \in \mathcal{C}$ such that if $\mathbf{v} = \frac{1}{2}\mathbf{v}_1 + \frac{1}{2}\mathbf{v}_2$, with $\mathbf{v}_1, \mathbf{v}_2 \in \mathcal{C}$, then $\mathbf{v} = \mathbf{v}_1 = \mathbf{v}_2$. Let us also define the support function map

$$\sigma_{\mathcal{Q}_{\mathbf{A}}}(\mathbf{v}) := \operatorname*{argmax}_{\mathbf{z} \in \mathcal{Q}_{\mathbf{A}}} \mathbf{v}^T\mathbf{z}. \tag{9}$$

Note that the maximum above is achieved at an extreme point of $\mathcal{Q}_{\mathbf{A}}$. For this reason, we refer $\sigma_{\mathcal{Q}_{\mathbf{A}}}(\mathbf{v})$ as the set of extreme points of $\mathcal{Q}_{\mathbf{A}}$ along $\mathbf{v}$. Note that $\sigma_{\mathcal{Q}_{\mathbf{A}}}(\mathbf{v})$ is not a singleton in general, but an exposed set. We remark that the endpoints of the spikes in Figure 3 and 4 are the extreme points in directions $\mathbf{e}_1$ and $\mathbf{e}_2$.

In the sequel, we show that the extreme points of $\mathcal{Q}_{\mathbf{A}}$ are given by data samples and convex mixtures of data samples in one dimensional and multidimensional cases, respectively. Here, we also provide a generic formulation for the extreme point along an arbitrary direction.

---

[5]We refer the reader to the supplementary material for the proof. For the remaining analysis, we drop the bias term, however, similar arguments also hold for a case with bias as illustrated in the supplementary file.

Lemma 2.6. *In a one dimensional data set ($d = 1$), for any vector $\mathbf{v} \in \mathbb{R}^n$, an extreme point of $\mathcal{Q}_\mathbf{A}$ along $\mathbf{v}$ is achieved when $u_v = \pm 1$ and $b_v = -sign(u_v)a_i$ for a certain index $i \in [n]$.*

Combined with Theorem 3.1, the above result proves that the optimal network outputs the linear spline interpolation for the input data. We now generalize the result for extreme points in the span of the ordinary basis vectors to higher dimensions. These will improve our spike-free relaxation as a first order correction. For instance, the behavior in Figure 3 and 4 is captured by the convex hull of the union of extreme points along $\mathbf{e}_1$ and $\mathbf{e}_2$, and the spike-free relaxation.

Lemma 2.7. *Extreme point in the span of each ordinary basis direction $\mathbf{e}_i$ is given by*

$$\mathbf{u}_i = \frac{\mathbf{a}_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} \lambda_j \mathbf{a}_j}{\left\| \mathbf{a}_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} \lambda_j \mathbf{a}_j \right\|_2} \quad and \quad b_i = \min_{j \neq i}(-\mathbf{a}_j^T \mathbf{u}_i), \tag{10}$$

where $\boldsymbol{\lambda}$ is computed via the following problem

$$\min_{\boldsymbol{\lambda}} \left\| \mathbf{a}_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} \lambda_j \mathbf{a}_j \right\|_2 \quad s.t. \ \boldsymbol{\lambda} \succcurlyeq \mathbf{0}, \mathbf{1}^T \boldsymbol{\lambda} = 1.$$

Lemma 2.7 shows that extreme points in the span of the ordinary basis directions yield neurons that can be represented as a convex mixture residual of the input data samples and the corresponding bias values can be computed via an inner product between the neurons and a data sample. Our next result characterizes extreme points along arbitrary directions for the general case.

Lemma 2.8. *For any $\boldsymbol{\alpha} \in \mathbb{R}^n$, the extreme point along the direction of $\boldsymbol{\alpha}$ can be found by*

$$\mathbf{u}_\alpha = \frac{\sum_{i \in \mathcal{S}}(\alpha_i + \lambda_i)\mathbf{a}_i - \sum_{j \in \mathcal{S}^c} \nu_j \mathbf{a}_j}{\| \sum_{i \in \mathcal{S}}(\alpha_i + \lambda_i)\mathbf{a}_i - \sum_{j \in \mathcal{S}^c} \nu_j \mathbf{a}_j \|_2} \quad and \quad b_\alpha = \begin{cases} \max_{i \in \mathcal{S}}(-\mathbf{a}_i^T \mathbf{u}_\alpha), & if \ \sum_{i \in \mathcal{S}} \alpha_i \leq 0 \\ \min_{j \in \mathcal{S}^c}(-\mathbf{a}_j^T \mathbf{u}_\alpha), & otherwise \end{cases} \tag{11}$$

where the set of active and inactive ReLUs, i.e., $\mathcal{S}$ and $\mathcal{S}^c$, $\boldsymbol{\lambda}$, and $\boldsymbol{\nu}$ are obtained via the following convex problem

$$\min_{\boldsymbol{\lambda}, \boldsymbol{\nu}} \max_{\mathbf{u}, b} \mathbf{u}^T \left(\sum_{i \in \mathcal{S}}(\alpha_i + \lambda_i)\mathbf{a}_i - \sum_{j \in \mathcal{S}^c} \nu_j \mathbf{a}_j \right) \quad s.t. \ \boldsymbol{\lambda}, \boldsymbol{\nu} \succcurlyeq 0, \sum_{i \in \mathcal{S}}(\alpha_i + \lambda_i) = \sum_{j \in \mathcal{S}^c} \nu_j, \|\mathbf{u}\|_2 \leq 1.$$

Lemma 2.8 proves that optimal neurons can be characterized as a linear combination of the input data samples. Below, we further simplify this characterization and obtain a representer theorem for regularized NNs.

Corollary 2.1 (A representer theorem for optimal neurons). *Lemma 2.8 implies that each extreme point along the direction $\boldsymbol{\alpha}$ can be written in the following compact form*

$$\mathbf{u}_\alpha = \frac{\sum_{i \in \mathcal{S}} \alpha_i(\mathbf{a}_i - \mathbf{a}_k)}{\| \sum_{i \in \mathcal{S}} \alpha_i(\mathbf{a}_i - \mathbf{a}_k) \|_2} \quad and \quad b = -\mathbf{a}_k^T \mathbf{u}_\alpha, \ for \ some \ k \ and \ subset \ \mathcal{S}.$$

Therefore, optimal neurons in the training objectives (2) and (4) all obey the above representation.

Remark 2.1. *An interpretation of the extreme points provided above is an auto-encoding of the data: the optimal neurons are convex mixture approximations of subsets of training samples via other subsets of training samples.*

3 Main results

In the following, we present our main findings based on the extreme point characterization introduced in the previous section.

3.1 Convex duality

In this section, we present our first duality result on the non-convex NN training objective given in (4).

Theorem 3.1. *The dual of the problem in* (4) *is given by*

$$D^* = \max_{\mathbf{v} \in \mathbb{R}^n} \mathbf{v}^T \mathbf{y} \qquad\qquad = \qquad\qquad \max_{\mathbf{v} \in \mathbb{R}^n} \mathbf{v}^T \mathbf{y} \,, \qquad (12)$$
$$s.t. \ \left| \mathbf{v}^T \left(\mathbf{A}\mathbf{u} \right)_+ \right| \leq 1 \, \forall \mathbf{u} \in \mathcal{B}_2 \qquad\qquad s.t. \ \mathbf{v} \in \mathcal{Q}_{\mathbf{A}}^\circ, -\mathbf{v} \in \mathcal{Q}_{\mathbf{A}}^\circ$$

and we have $P^* \geq D^*$. *For finite width NNs, there exists a large enough m such that we have strong duality, i.e., $P^* = D^*$, and an optimal $\mathbf{U}$ for* (4) *satisfies* $\|(\mathbf{A}\mathbf{U}^*)_+^T \mathbf{v}^*\|_\infty = 1$, *where $\mathbf{v}^*$ is dual optimal.*[6]

Remark 3.1. *Note that* (12) *is a convex optimization problem with infinitely many constraints, and in general not polynomial-time tractable. In fact, even checking whether a point $\mathbf{v}$ is feasible is NP-hard: we need to solve* $\max_{\mathbf{u}: \|\mathbf{u}\|_2 \leq 1} \sum_{i=1}^n v_i \left(\mathbf{a}_i^T \mathbf{u} \right)_+$. *This is related to the problem of learning halfspaces with noise, which is NP-hard to approximate within a constant factor (see e.g. [15, 7]).*

Based on the dual form and the optimality condition in Theorem 3.1, we can characterize the optimal neurons as the extreme points of a certain set.

Corollary 3.1. *Theorem 3.1 implies that the optimal neuron weights are extreme points which solve the following optimization problem*

$$\operatorname*{argmax}_{\mathbf{u}: \|\mathbf{u}\|_2 \leq 1} |\mathbf{v}^{*T} \left(\mathbf{A}\mathbf{u} \right)_+ |.$$

The above corollary shows that the optimal neuron weights are extreme points along $\pm \mathbf{v}^*$ given by $\sigma_{\mathcal{Q}_{\mathbf{A}}}(\pm \mathbf{v}^*)$ for some dual optimal parameter $\mathbf{v}^*$.

In the sequel, we provide a theoretical analysis for the duality gap of finite width NNs.

3.1.1 Duality for finite width neural networks

The following theorem proves that weak duality holds for any finite width NN.

Theorem 3.2. *Suppose that* (4) *is feasible, then weak duality holds for* (4).

We now prove that when the number of extreme points is finite, i.e., we have finitely many possible optimal neuron weights, strong duality holds for any finite width NN.

Theorem 3.3. *Let $\{\mathbf{A}, \mathbf{y}\}$ be a dataset such that the number of solutions to Corollary 3.1 is finite and* (4) *is feasible. Then, strong duality holds for* (4).

Since strong duality holds for finite width NNs as proved in Theorem 3.3, we can achieve the minimum of the primal problem in (4) through the dual form in (12). Therefore, the NN architecture in (1) can be globally optimized via a subset of extreme points defined in Corollary 3.1.

In the sequel, we first show that there exist finitely many extreme points for some specific problems. We then prove that strong duality holds for these problems.

---

[6]Similar results hold for vector output networks. We defer these results to the supplementary file and present our results in this simplified version.
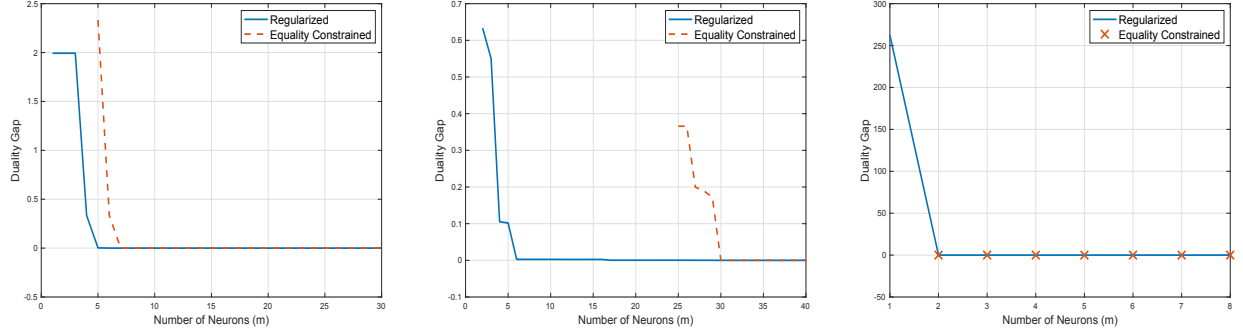
(a) Duality gap for the one dimensional dataset in Figure 1b.

(b) Duality gap for a rank-one dataset with $n = 15$ and $d = 10$.

(c) Duality gap for a whitened dataset with $n = 30$ and $d = 40$.

Figure 6: Duality gap for a regression scenario, where we select $\beta = 10^{-3}$ for the regularized problem. Here, we consider both the equality constrained case, i.e., (4), and the regularized case, i.e., (16).

3.2 Structure of one dimensional networks

We are now ready to present our results on the structure induced by the extreme points for one dimensional problems. The following corollary directly follows from Lemma 2.6.

Corollary 3.2. *Let $\{a_i\}_{i=1}^n$ be a one dimensional training set i.e., $a_i \in \mathbb{R}$, $\forall i \in [n]$. Then, a set of solutions to (4) that achieve the optimal value are extreme points, and therefore satisfy $\{(u_i, b_i)\}_{i=1}^m$, where $u_i = \pm 1, b_i = -sign(u_i)a_i$.*

In Figure 6a, we perform a numerical experiment on the dataset plotted in Figure 1b. Since this case has finitely many extreme points as proved in Corollary 3.2, duality gap vanishes and strong duality holds when we reach a certain m value. Notice that this result also validates Theorem 3.3.

Corollary 3.3. *For problems involving one dimensional training data, strong duality holds as a result of Corollary 3.2 and Theorem 3.3.*

As a result of Corollary 3.3, we can globally optimize (1) using a subset of finite number of solutions in Corollary 3.2. In addition to this, in the following, we prove that there might exist multiple optimal solutions to (4).

Proposition 3.1. *The solution provided in Corollary 3.2 may not be unique. In the supplementary file, we present a counter-example where an optimal solution is not in this form, i.e., not a piecewise linear spline.*

3.3 Solutions to rank-one problems

In this section, we first characterize all possible extreme points for problems involving rank-one data matrices. We then prove that strong duality holds for these problems.

Corollary 3.4. *Let $\mathbf{A}$ be a data matrix such that $\mathbf{A} = \mathbf{c}\mathbf{a}^T$, where $\mathbf{c} \in \mathbb{R}^n$ and $\mathbf{a} \in \mathbb{R}^d$. Then, a set of solutions to (4) that achieve the optimal value are extreme points, and therefore satisfy $\{(\mathbf{u}_i, b_i)\}_{i=1}^m$, where $\mathbf{u}_i = s_i \frac{\mathbf{a}}{\|\mathbf{a}\|_2}, b_i = -s_i c_i \|\mathbf{a}\|_2$ with $s_i = \pm 1, \forall i \in [m]$.*

11

Corollary 3.5. *As a result of Theorem 3.3 and Corollary 3.4, strong duality holds for problems involving rank-one data matrices.*

Corollary 3.4 and 3.5 indicate that, as in the previous section, rank-one problems have finite number of solutions to (4) so that we can globally optimize regularized NNs using a subset of these solutions. We also present a numerical example in Figure 6b to confirm the theoretical prediction of Corollary 3.5.

3.4 Closed form solutions and ℓ_0-ℓ_1 equivalence

A considerable amount of literature have been published on the equivalence of minimal ℓ_1 and ℓ_0 solutions in under-determined linear systems, where it was shown that the equivalence holds under assumptions on the data matrices (see e.g. [16, 17, 18]). We now prove a similar equivalence for two-layer NNs. Consider the minimal cardinality problem

$$\min_{\theta \in \Theta} \|\mathbf{w}\|_0 \text{ s.t. } f_\theta(\mathbf{A}) = \mathbf{y}, \|\mathbf{u}_j\|_2 = 1, \forall j. \tag{13}$$

The following results provide a characterization of the optimal solutions to the above problem

Lemma 3.1. *Suppose that $n \leq d$, $\mathbf{A}$ is full row rank and $\mathbf{y}$ contains both positive and negative entries, and define $\mathbf{A}^\dagger = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}$. Then an optimal solution to the problem in (13) is given by*

$$\mathbf{u}_1 = \frac{\mathbf{A}^\dagger(\mathbf{y})_+}{\|\mathbf{A}^\dagger(\mathbf{y})_+\|_2} , \ w_1 = \|\mathbf{A}^\dagger(\mathbf{y})_+\|_2 \quad and \quad \mathbf{u}_2 = \frac{\mathbf{A}^\dagger(-\mathbf{y})_+}{\|\mathbf{A}^\dagger(-\mathbf{y})_+\|_2}, \ w_2 = -\|\mathbf{A}^\dagger(-\mathbf{y})_+\|_2.$$

Lemma 3.2. *We have ℓ_1-ℓ_0 equivalence, i.e., the optimal solutions of (13) and (4) coincide if the following condition holds*

$$\min_{\mathbf{v}:\mathbf{v}^T(\mathbf{A}\mathbf{u}_1)_+=1, \mathbf{v}^T(\mathbf{A}\mathbf{u}_2)_+=-1} \quad \max_{\mathbf{u}:\|\mathbf{u}\|_2 \leq 1} \left|\mathbf{v}^T(\mathbf{A}\mathbf{u})_+\right| \leq 1.$$

Furthermore, whitened data matrices with $n \leq d$ satisfy ℓ_1-ℓ_0 equivalence.

Corollary 3.6. *As a result of Theorem 3.3 and Corollary 3.2, strong duality holds for problems involving whitened data matrices.*

Lemma 3.2 and Corollary 3.6 prove that (1) can be globally optimized using a subset of extreme points. We also validate Corollary 3.6 via a numerical experiment in Figure 6c.

3.5 A cutting plane method

In this section, we introduce a cutting plane based training algorithm for the NN in (1). Among infinitely many possible unit norm weights, we need to find the weights that violate the inequality constraint in (12), which can be done by solving the following optimization problems

$$\mathbf{u}_1^* = \operatorname*{argmax}_{\mathbf{u}:\|\mathbf{u}\|_2 \leq 1} \mathbf{v}^T(\mathbf{A}\mathbf{u})_+ \qquad \mathbf{u}_2^* = \operatorname*{argmin}_{\mathbf{u}:\|\mathbf{u}\|_2 \leq 1} \mathbf{v}^T(\mathbf{A}\mathbf{u})_+. \tag{14}$$

However, (14) is not a convex problem since ReLU is a convex function. There exist several methods and relaxations to find the optimal parameters for (14). As an example, one can use the Frank-Wolfe algorithm [19] in order to approximate the solution iteratively. In the following, we show how to relax the problem using our spike-free relaxation

$$\hat{\mathbf{u}}_1 = \underset{\mathbf{u}:\mathbf{Au}\succcurlyeq\mathbf{0},\|\mathbf{u}\|_2\leq 1}{\operatorname{argmax}} \mathbf{v}^T\mathbf{Au} \qquad \hat{\mathbf{u}}_2 = \underset{\mathbf{u}:\mathbf{Au}\succcurlyeq\mathbf{0},\|\mathbf{u}\|_2\leq 1}{\operatorname{argmin}} \mathbf{v}^T\mathbf{Au}, \tag{15}$$

where we relax the set $\{(\mathbf{Au})_+|\mathbf{u} \in \mathbb{R}^d, \|\mathbf{u}\|_2 \leq 1\}$ as $\{\mathbf{Au}|\mathbf{u} \in \mathbb{R}^d, \|\mathbf{u}\|_2 \leq 1\} \cap \mathbb{R}^n_+$. Now, we can find the weights for the hidden layer using (15). In the cutting plane method, we first find a violating neuron using (15). After adding these parameters to $\mathbf{U}$ as columns, we solve (4). If we cannot find a new violating neuron then we terminate the algorithm. Otherwise, we find the dual parameter for the updated $\mathbf{U}$. We repeat this procedure till we find an optimal solution. We also provide the full algorithm in Algorithm 1[7].

Algorithm 1 Cutting Plane based Training Algorithm for Two-Layer NNs (without bias)

1: Initialize $\mathbf{v} = \mathbf{y}$
2: **while** there exists a violating neuron **do**
3:     Find $\hat{\mathbf{u}}_1$ and $\hat{\mathbf{u}}_2$ via (15)
4:     $\mathbf{U} \leftarrow [\mathbf{U}\ \hat{\mathbf{u}}_1\ \hat{\mathbf{u}}_2]$
5:     Find $\mathbf{v}$ using the dual problem in (12)
6:     Check the existence of a violating neuron via (15)
7: **end while**
8: Solve (4) using $\mathbf{U}$
9: Return $\theta = (\mathbf{U}, \mathbf{w})$

Proposition 3.2. *When $\mathbf{A}$ is spike-free as defined in Lemma 2.3, the cutting plane based training method globally optimizes* (12).

The following theorem shows that random high dimensional i.i.d. Gaussian matrices asymptotically satisfy the spike-free condition.

Theorem 3.4. *Let $\mathbf{A} \in \mathbb{R}^{n\times d}$ be an i.i.d. Gaussian random matrix. Then $\mathbf{A}$ is asymptotically spike-free as $d \to \infty$. More precisely, we have*

$$\lim_{d\to\infty} \mathbb{P}\left[\max_{\mathbf{u}\in\mathcal{B}_2}\|\mathbf{A}^\dagger(\mathbf{Au})_+\|_2 > 1\right] = 0\,.$$

We now consider improving the basic relaxation by including the extreme points in our relaxation, and provide some theoretical results.

Theorem 3.5. *Let $\mathcal{C}_a$ denote the convex hull of $\{\mathbf{a}_i\}_{i=1}^n$. If each sample is a vertex of $\mathcal{C}_a$, then a feasible solution to (4) can be achieved with n neurons, which are the extreme points in the span of the ordinary basis vectors. Consequently, the weights given in Lemma 2.7 achieve zero training error.*

Our next result shows that the above condition is likely to hold for high dimensional random matrices.

Theorem 3.6. *Let $\mathbf{A} \in \mathbb{R}^{n\times d}$ be a data matrix generated i.i.d. from a standard Gaussian distribution $\mathcal{N}(0,1)$. Suppose that the dimensions of the data matrix obey $d > 2n\log(n-1)$. Then, every row of $\mathbf{A}$ is an extreme point of the convex hull of the rows of $\mathbf{A}$ with high probability.*

---

[7]We also provide a cutting plane method with bias in the supplementary material.

4 Regularized two-layer ReLU networks

Here, we first formulate a penalized version of the equality form in (4). We then present duality results for this case.

Theorem 4.1. *An optimal* $\mathbf{U}$ *for the following regularized version of* (4) *given by*

$$\min_{\theta \in \Theta} \frac{1}{2} \|(\mathbf{AU})_+ \mathbf{w} - \mathbf{y}\|_2^2 + \beta \|\mathbf{w}\|_1 \ \ s.t. \ \|\mathbf{u}_j\|_2 \le 1, \forall j, \tag{16}$$

can be found through the following dual problem

$$\max_{\mathbf{v}} -\frac{1}{2} \|\mathbf{v} - \mathbf{y}\|_2^2 + \frac{1}{2} \|\mathbf{y}\|_2^2 \ \ s.t. \ \mathbf{v} \in \beta \mathcal{Q}_{\mathbf{A}}^\circ, -\mathbf{v} \in \beta \mathcal{Q}_{\mathbf{A}}^\circ,$$

where β *is the regularization (weight decay) parameter.*

Remark 4.1. *We note that all the weak and strong duality results in Theorem 3.2 and 3.3 hold for regularized networks. Therefore, Corollary 3.3, 3.5 and 3.6 apply to regularized networks. The numerical results in Figure 6 also confirm this claim.*

Corollary 4.1. *Remark 4.1 also implies that whenever there exist finitely many extreme points, e.g., problems involving rank-one and/or whitened data matrices, we can solve* (4) *as a convex* ℓ_1*-norm minimization problem to achieve the optimal solutions. Particularly, we first construct a hidden layer weight matrix, i.e., denoted as* $\mathbf{U}^*$*, using all possible extreme points. We then solve the following problem*

$$\min_{\mathbf{w}} \|\mathbf{w}\|_1 \ \ s.t. \ \left(\mathbf{AU}^*\right)_+ \mathbf{w} = \mathbf{y} \tag{17}$$

or the corresponding regularized version

$$\min_{\mathbf{w}} \frac{1}{2} \|\left(\mathbf{AU}^*\right)_+ \mathbf{w} - \mathbf{y}\|_2^2 + \beta \|\mathbf{w}\|_1.$$

5 Two-layer ReLU networks with hinge loss

Now we consider the classification problem with the hinge loss.

Theorem 5.1. *An optimal* $\mathbf{U}$ *for the binary classification task with the hinge loss given by*

$$\min_{\theta \in \Theta} \sum_{i=1}^n \max\{0, 1 - y_i(\mathbf{a}_i^T \mathbf{U})_+ \mathbf{w}\} + \beta \|\mathbf{w}\|_1 \ \ s.t. \ \|\mathbf{u}_j\|_2 \le 1, \forall j, \tag{18}$$
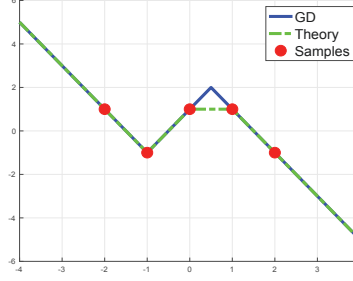
can be found through the following dual

$$\max_{\mathbf{v}} \mathbf{v}^T \mathbf{y} \ \ s.t. \ 0 \le y_i v_i \le 1 \ \forall i \in [n], \mathbf{v} \in \beta \mathcal{Q}_{\mathbf{A}}^\circ, -\mathbf{v} \in \beta \mathcal{Q}_{\mathbf{A}}^\circ.$$
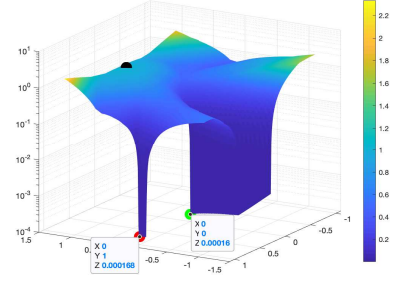
Theorem 5.1 proves that since strong duality holds for two-layer wide NNs, we can obtain the optimal solutions to (18) through the dual form. The following corollary characterizes the solutions obtained via the dual form of (18).
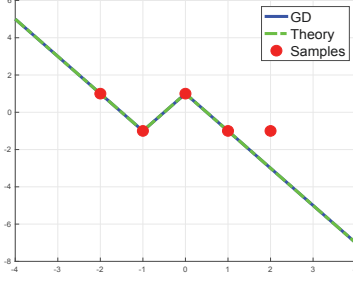
(a) $L_t = 1.600 \times 10^{-4}$ and $L_{gd} = 1.600 \times 10^{-4}$. GD and our theory agrees.

(b) $L_t = 1.600 \times 10^{-4}$ and $L_{gd} = 1.679 \times 10^{-4}$. GD is stuck at a local minimum.

(c) Visualization of the loss landscape in (b) ($L_t = 1.600 \times 10^{-4}$ and $L_{gd} = 1.679 \times 10^{-4}$).

Figure 7: Binary classification using hinge loss, where we apply GD and our approach in Theorem 5.2. Here, we denote the objective value in (18) as L_t and L_{gd} for our theoretical approach (Theorem 5.2) and GD, respectively. In (c), we also provide 3D illustration of the loss surface of the example in (b), where we mark the initial point (black), the GD solution (red), and our solution (green).

Corollary 5.1. *Theorem 4.1 implies that the optimal neuron weights are extreme points which solve the following optimization problem*

$$\operatorname*{argmax}_{\mathbf{u}:\|\mathbf{u}\|_2 \le 1} |\mathbf{v}^{*T}\big(\mathbf{A}\mathbf{u}\big)_+|,$$

where $\|\mathbf{v}^*\|_\infty \le 1$.

Consequently, in the 1D case, the optimal neuron weights are given by the extreme points as a result of Theorem 3.1. Therefore the optimal network network output is given by the piecewise linear function

$$f(\mathbf{a}) = \sum_{j=1}^{m} w_j(\mathbf{a}u_j + b_j)_+ \,,$$

for some output weights $w_1, \ldots, w_m$ where $u_j = \pm 1$ and $b_j = \mp a_j$ for some j.

This explains Figure 1c, where the decision regions are determined by the zero crossings of the above piecewise linear function. Moreover, the dual problem reduces to a finite dimensional minimum ℓ_1 norm Support Vector Machine (SVM), whose solution can be easily determined. As it can be seen in Figure 1c, the piecewise linear fit passes through the data samples which are on the margin, i.e., the network output is ± 1. This corresponds to the maximum margin decision regions and separates the green shaded area from the red shaded area.

It is straightforward to see that this is equivalent to applying the kernel map $\kappa(a, a_j) = (a - a_j)_+$, forming the corresponding kernel matrix

$$\mathbf{K}_{ij} = (a_i - a_j)_+ \,,$$

and solving minimum ℓ_1-norm SVM on the kernelized data matrix. The same steps can also be applied to a rank-one dataset as presented in the following.

15

Corollary 5.2. *Let* $\mathbf{A}$ *be a data matrix such that* $\mathbf{A} = \mathbf{c}\mathbf{a}^T$, *where* $\mathbf{c} \in \mathbb{R}^n$ *and* $\mathbf{a} \in \mathbb{R}^d$. *Then, a set of solutions to* (18) *that achieve the optimal value are extreme points, and therefore satisfy* $\{(\mathbf{u}_i, b_i)\}_{i=1}^m$, *where* $\mathbf{u}_i = s_i \frac{\mathbf{a}}{\|\mathbf{a}\|_2}$, $b_i = -s_i c_i \|\mathbf{a}\|_2$ *with* $s_i = \pm 1, \forall i \in [m]$.

The proof directly follows from the proof of Corollary 3.4.

Theorem 5.2. *For a rank-one dataset* $\mathbf{A} = \mathbf{c}\mathbf{a}^T$, *applying* ℓ_1-*norm SVM on* $\left(\mathbf{A}\mathbf{U}^{*T} + \mathbf{1}\mathbf{b}^{*T}\right)_+$ *finds the optimal solution* θ^* *to* (18), *where* $\mathbf{U}^* \in \mathbb{R}^{d \times 2n}$ *and* $\mathbf{b}^* \in \mathbb{R}^{2n}$ *are defined as* $\{\mathbf{u}_i^* = s_i \frac{\mathbf{a}}{\|\mathbf{a}\|_2}, b_i^* = -s_i c_i \|\mathbf{a}\|_2\}_{i=1}^n$ *with* $s_i = \pm 1, \forall i$.

The proof direclty follows from Corollary 5.2. We also verify Theorem 5.2 using a one dimensional dataset in Figure 7. In this figure, we observe that whenever there is a sign change, the corresponding two samples determine the decision boundary, which resembles the idea of support vector. Thus, the piecewise linear fit passes through these samples. On the other hand, when there is no sign change, the piecewise fit does not create any kink as in Figure 7a. We also observe that GD might fail to globally optimize (18) unlike our approach as illustrated in Figure 7b. In Figure 7c, we also provide a visualization of the loss landscape for this case.

6 Two-layer ReLU networks with general loss functions

Now we consider the scalar output two-layer ReLU networks with an arbitrary convex loss function

$$\min_{\theta \in \Theta} \ell((\mathbf{A}\mathbf{U})_+ \mathbf{w}, \mathbf{y}) + \beta \|\mathbf{w}\|_1 \text{ s.t. } \|\mathbf{u}_j\|_2 \leq 1, \forall j, \tag{19}$$

where $\ell(\cdot, \mathbf{y})$ is a convex loss function.

Theorem 6.1. *The dual of* (19) *is given by*

$$\max_{\mathbf{v}} -\ell^*(\mathbf{v}) \text{ s.t. } \mathbf{v} \in \beta \mathcal{Q}_{\mathbf{A}}^\circ, -\mathbf{v} \in \beta \mathcal{Q}_{\mathbf{A}}^\circ,$$

where ℓ^* *is the Fenchel conjugate function defined as*

$$\ell^*(\mathbf{v}) = \max_{\mathbf{z}} \mathbf{z}^T \mathbf{v} - \ell(\mathbf{z}, \mathbf{y}).$$

Theorem 6.1 proves that our extreme point characterization in Lemma 2.6 applies to arbitrary convex loss function. Therefore, the optimal parameters for (1) is a subset of the same extreme point set, i.e., determined by the input data matrix $\mathbf{A}$, independent of loss function.

7 Comparison with Neural Tangent Kernel

Here, we first briefly discuss the recently introduced Neural Tangent Kernel [9] and other connections to kernel methods. We then compare this approach with our exact characterization in terms of a kernel matrix in Corollary 4.1.

The connection between kernel methods and infinitely wide NNs has been extensively studied [20, 21, 22]. Earlier studies typically considered untrained networks, or the training of the last layer while keeping the hidden layers fixed and random. Then, by assuming a distribution for initialization of the parameters, the behavior of an infinitely wide NN can be captured by the kernel matrix $\mathbf{K}(\mathbf{a}_i, \mathbf{a}_j) = \mathbb{E}_{\theta \sim \mathcal{D}}[f_\theta(\mathbf{a}_i) f_\theta(\mathbf{a}_j)]$, where $\mathcal{D}$ is the distribution for initialization. However, these results do not fully align with practical NNs where all the layers are trained. Consequently,
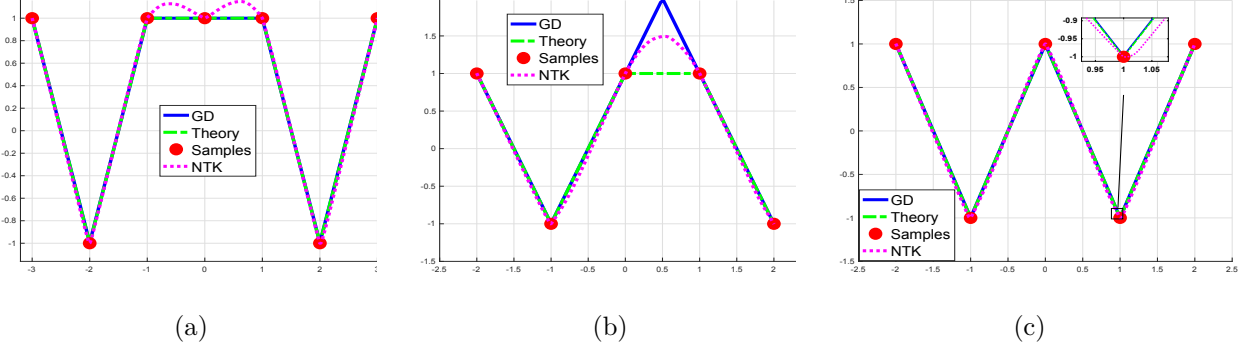
Figure 8: One dimensional regression task with square loss, where we apply GD, NTK, and our approach in Corollary 4.1.

the authors in [9] introduced a new kernel, i.e., NTK, where all the layers are trained while the width tends to infinity. In this scenario, the network can be characterized by the kernel matrix $\mathbf{K}(\mathbf{a}_i, \mathbf{a}_j) = \mathbb{E}_{\theta \sim \mathcal{D}}[\nabla_\theta f_\theta(\mathbf{a}_i)^T \nabla_\theta f_\theta(\mathbf{a}_j)]$. This can be seen as a linearization of the NN model under a particular scaling assumption, and is closely related to random feature methods. We refer the reader to [8] for details and limitations of the NTK framework. For one dimensional problems, this kernel characterization can be written as follows [23][8]

$$\mathbf{K}(a_i, a_j) = |a_i||a_j|\kappa\left(\frac{a_i a_j}{|a_i||a_j|}\right),$$

where

$$\kappa(u) = u\kappa_0(u) + \kappa_1(u)$$

$$\kappa_0(u) = \frac{1}{\pi}(\pi - \arccos(u)) \ \kappa_1(u) = \frac{1}{\pi}\left(u(\pi - \arccos(u)) + \sqrt{1-u^2}\right).$$

After forming the kernel matrix, one can solve the following ℓ_2-norm minimization problem to obtain the last layer weights

$$\min \|\mathbf{w}\|_2^2 \text{ s.t. } \mathbf{K}\mathbf{w} = \mathbf{y}. \tag{20}$$

We first note that our approach in (17) is different than the NTK approach in (20) in terms of kernel construction and objective function.

In order to compare the performance of (17), (20) and GD, we perform experiments on one dimensional datasets. In Figure 8, we observe that NTK outputs smoother functions compared to GD and our approach. Particularly, in Figure 8a and 8b, we clearly see that the output of NTK is not a piecewise linear function unlike our approach and GD. Moreover, even though the output of NTK looks like a piecewise linear function in Figure 8c, we again observe its smooth behavior around the data points. Thus, we conclude that NTK yields output functions that are significantly different than the piecewise linear functions obtained by GD and our approach. We also want to emphasize that although our solution and GD yield different output functions in Figure 8b, the regularized objective value of our solution is smaller as shown in Figure 7b. The reason is that GD can be stuck at local minima even in simple one dimensional regression problems. Furthermore, GD is also able to achieve the solution provided by our approach in some of our trials, when initialized in the basin of attraction that contains the optimal solution.

---

[8]We provide the NTK formulation without a bias term to simplify the presentation. The expression for the case with bias can be found in [24].
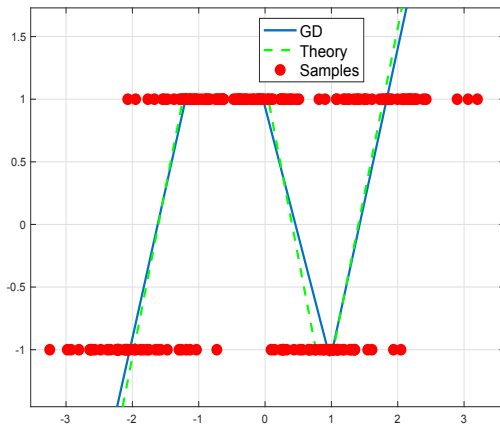
(a) Visualization of the samples and the function fit.
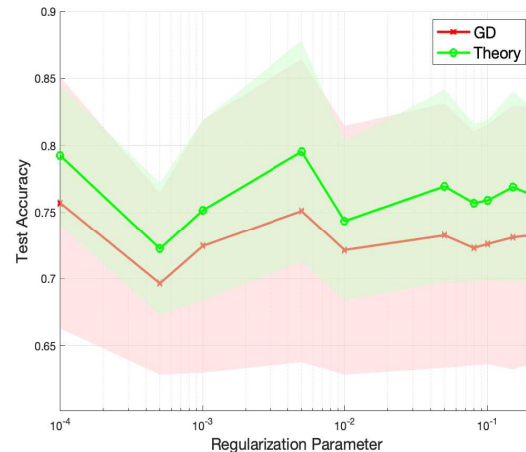
(b) Test accuracy with 90% confidence band.

Figure 9: Binary classification using hinge loss, where we apply GD and our approach in Theorem 5.2, i.e., denoted as Theory.

Table 1: Classification Accuracies (%) and test errors

	MNIST	CIFAR-10	Bank	Boston	California	Elevators	News20	Stock
One Layer NN (Least Squares)	86.04%	36.39%	0.9258	0.3490	0.8158	0.5793	1.0000	1.0697
Two-Layer NN (Backpropagation)	96.25%	41.57 %	0.6440	0.1612	0.8101	0.4021	0.8304	0.8684
Two-Layer NN Convex	96.94%	42.16%	**0.5534**	**0.1492**	**0.6344**	**0.3757**	**0.8043**	**0.6184**
Two-Layer Convex-RF	**97.72**%	**80.28**%	-	-	-	-	-	-

8   Numerical experiments

We first consider a binary classification experiment using hinge loss on a synthetic dataset. To generate a dataset, we use a Gaussian mixture model, i.e., $a_i \sim \mathcal{N}(\mu_j, 0.25)$, where the labels are computed using: $y_i = 1$, if $\mu_j \in \{-1, 0, 2\}$, and $y_i = -1$, if $\mu_j \in \{-2, 1\}$. Following these steps, we generate multiple datasets with nonoverlapping training and test splits. We then run our approach in Theorem 5.2, i.e., denoted as Theory and GD on these datasets. In Figure 9, we plot the mean test accuracy (solid lines) of each algorithm along with a one standard deviation confidence band (shaded regions). As illustrated in this example, our approach achieves slightly better generalization performance compared to GD. We also visualize the sample data distributions and the corresponding function fits in Figure 9a, where we provide an example to show the agreement between the solutions found by our approach and GD.

We then consider classification tasks and report the performance of the algorithms on MNIST [25] and CIFAR-10 [26][9]. For these data sets, we do not perform any preprocessing except a normalization on the pixels in MNIST so that each pixel is in the range $[0, 1]$. In Table 1, we observe that our approach denoted as `Convex`, which based on convex optimization, outperforms the non-convex backpropagation based approach. In addition, we use an alternative approach,

---

[9]We use a generalized vector output version of our method discussed in the supplementary material, where further details on the numerical experiments are available.

Figure 10: Extreme points found by (10) applied on image patches of CIFAR-10 yield filters used in the `Convex-RF` algorithm. In this example, we don't apply the whitening step and observe that filters resemble actual image patches.

denoted as `Convex-RF` in Table 1 which uses (10) on image patches to obtain filters, e.g., Figure 10. This training approach for the hidden layer weights surprisingly increases the accuracy by almost 40% compared to the convex approach with the cutting plane algorithm. We also evaluate the performances on several regression data sets, namely Bank, Boston Housing, California Housing, Elevators, Stock [27], and the Twenty Newsgroups text classification data set [28]. In Table 1, we provide the test errors for each approach. Here, our convex approach outperforms the backpropagation, and the one layer NN approach in each case.

8.1 Unsupervised training approach for the hidden layer weights (`Convex-RF`)

Algorithm 2 `Convex-RF`

1: Set P, ϵ, and β
2: Randomly select P patches from the dataset: $\{\mathbf{p}_i\}_{i=1}^{P}$
3: **for** i=1:P **do**
4:     Normalize the patch: $\bar{\mathbf{p}}_i = \frac{\mathbf{p}_i - \text{mean}(\mathbf{p}_i)}{\sqrt{\text{var}(\mathbf{p}_i) + \epsilon}}$
5: **end for**
6: Form a patch matrix $\mathbf{P} = [\bar{\mathbf{p}}_1 \ldots \bar{\mathbf{p}}_P]$
7: (Optional) Apply whitening to the patch matrix:

$$[\mathbf{V}, \mathbf{D}] = \text{eig}(\text{cov}(\mathbf{P}))$$
$$\tilde{\mathbf{P}} = \mathbf{V}(\mathbf{D} + \epsilon\mathbf{I})^{-\frac{1}{2}}\mathbf{V}^T\mathbf{P}$$

8: **for** i=1:P **do**
9:     Compute a neuron using (10):

$$\mathbf{u}_i = \frac{\tilde{\mathbf{p}}_i - \sum_{\substack{j=1 \\ j\neq i}}^{n} \lambda_j \tilde{\mathbf{p}}_j}{\left\| \tilde{\mathbf{p}}_i - \sum_{\substack{j=1 \\ j\neq i}}^{n} \lambda_j \tilde{\mathbf{p}}_j \right\|_2}$$

10: **end for**
11: Form the neuron matrix: $\mathbf{U} = [\mathbf{u}_1 \ldots \mathbf{u}_P]$
12: Extract all the patches in $\mathbf{A}$: $\mathbf{A}_p$
13: Compute activations: $\mathbf{B} = \text{pooling}(\text{ReLU}(\mathbf{A}_p\mathbf{U}))$
14: Solve a convex ℓ_1-norm minimization problem: $\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{B}\mathbf{w} - \mathbf{y}\|_2^2 + \beta\|\mathbf{w}\|_1$

For this approach, we use the convolutional neural net architecture in [29]. However, instead of using random filters as in [4] or applying the k-means algorithm as in [29], we use the filters that are

extracted from the patches using our convex approach in (10). Particularly, we first randomly obtain patches from the dataset. We then normalize and whiten (using the ZCA whitening approach) the randomly selected patches. After that we apply (10) on the the resulting patches to obtain the filter weights via a convex optimization problem with the unit simplex constraints.

After obtaining the filter weights in an unsupervised manner, we first compute the activations for each input patch. Here, we use a linear function for activations unlike the triangular activation function in [29]. We then apply ReLU and max pooling. Finally, we solve a convex ℓ_1-norm minimization problem to obtain the output layer weights. Therefore, we achieve a training approach that completely utilizes convex optimization tools and learns the hidden layer weights in an unsupervised manner. The complete algorithm is also presented in Algorithm 2.

9   Concluding remarks

We have studied two-layer ReLU networks via a convex analytic framework that explains why simple solutions are achieved even when networks are over-parameterized. In particular, we showed that the extreme points characterize simple structures and explain why training of regularized NNs yields a linear spline interpolation in 1D. We also showed that a finite set of optimal solutions for the hidden layer can be obtained via our extreme point characterization in certain cases. Thus, the original training problem can be simplified to a simple convex optimization problem with linear constraints. Using these observations, we also provided a training algorithm based on cutting planes, which achieves global optimality under certain assumptions. Additionally, based on the proven equivalence of $\ell_1 - \ell_2$ regularized training problems, we established a relation with minimum cardinality problems in compressed sensing. This relation allowed us to obtain closed form solutions for the network parameters in some cases. Therefore, our current results precisely characterize the structure of the set of globally optimal solutions, consequently, prove nonuniqueness. Based on our preliminary results on two-layer ReLU networks, we conjecture that similar extreme point characterizations in deep ReLU networks may explain their extraordinary generalization properties in practice.

References

[1] Hartmut Maennel, Olivier Bousquet, and Sylvain Gelly. Gradient descent quantizes relu network features. *arXiv preprint arXiv:1803.08367*, 2018.

[2] Pedro Savarese, Itay Evron, Daniel Soudry, and Nathan Srebro. How do infinite width bounded norm networks look in function space? *CoRR*, abs/1902.05040, 2019.

[3] Guy Blanc, Neha Gupta, Gregory Valiant, and Paul Valiant. Implicit regularization for deep neural networks driven by an ornstein-uhlenbeck like process. *CoRR*, abs/1904.09080, 2019.

[4] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

[5] Yoshua Bengio, Nicolas L Roux, Pascal Vincent, Olivier Delalleau, and Patrice Marcotte. Convex neural networks. In *Advances in neural information processing systems*, pages 123–130, 2006.

[6] Colin Wei, Jason D Lee, Qiang Liu, and Tengyu Ma. On the margin theory of feedforward neural networks. *arXiv preprint arXiv:1810.05369*, 2018.

[7] Francis Bach. Breaking the curse of dimensionality with convex neural networks. *The Journal of Machine Learning Research*, 18(1):629–681, 2017.

[8] Lenaic Chizat and Francis Bach. A note on lazy training in supervised differentiable programming. *arXiv preprint arXiv:1812.07956*, 2018.

[9] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pages 8571–8580, 2018.

[10] Sanjeev Arora, Simon S. Du, Wei Hu, Zhiyuan Li, Ruslan Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. *CoRR*, abs/1904.11955, 2019.

[11] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*, 2014.

[12] Rahul Parhi and Robert D Nowak. Minimum "norm" neural networks are splines. *arXiv preprint arXiv:1910.02333*, 2019.

[13] Peng Zhao and Bin Yu. On model selection consistency of lasso. *Journal of Machine learning research*, 7(Nov):2541–2563, 2006.

[14] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, 1970.

[15] Venkatesan Guruswami and Prasad Raghavendra. Hardness of learning halfspaces with noise. *SIAM Journal on Computing*, 39(2):742–765, 2009.

[16] E. J. Candes and T. Tao. Decoding by linear programming. *IEEE Trans. Info Theory*, 51(12):4203–4215, December 2005.

[17] D. L. Donoho. For most large underdetermined systems of linear equations, the minimal ℓ_1-norm solution is also the sparsest solution. *Communications on Pure and Applied Mathematics*, 59(6):797–829, June 2006.

[18] GM Fung and OL Mangasarian. Equivalence of minimal ℓ_0-and ℓ_p-norm solutions of linear equalities, inequalities and linear programs for sufficiently small p. *Journal of optimization theory and applications*, 151(1):1–10, 2011.

[19] Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(12):95–110, 1956.

[20] Radford M Neal. Priors for infinite networks. In *Bayesian Learning for Neural Networks*, pages 29–53. Springer, 1996.

[21] Alexander G de G Matthews, Mark Rowland, Jiri Hron, Richard E Turner, and Zoubin Ghahramani. Gaussian process behaviour in wide deep neural networks. *arXiv preprint arXiv:1804.11271*, 2018.

[22] Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. *arXiv preprint arXiv:1711.00165*, 2017.

[23] Alberto Bietti and Julien Mairal. On the inductive bias of neural tangent kernels. *arXiv preprint arXiv:1905.12173*, 2019.

[24] Francis Williams, Matthew Trager, Claudio Silva, Daniele Panozzo, Denis Zorin, and Joan Bruna. Gradient dynamics of shallow univariate relu networks. *arXiv preprint arXiv:1906.07842*, 2019.

[25] Yann LeCun. The MNIST database of handwritten digits. `http://yann.lecun.com/exdb/mnist/`.

[26] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. The CIFAR-10 dataset. `http://www.cs.toronto.edu/kriz/cifar.html`, 2014.

[27] L. Torgo. Regression data sets. `http://www.dcc.fc.up.pt/~ltorgo/Regression/DataSets.html`.

[28] Tom M Mitchell and Machine Learning. Mcgraw-hill science. *Engineering/Math*, 1:27, 1997.

[29] Adam Coates and Andrew Y Ng. Learning feature representations with k-means. In *Neural networks: Tricks of the trade*, pages 561–580. Springer, 2012.

[30] Anders Krogh and John A Hertz. A simple weight decay can improve generalization. In *Advances in neural information processing systems*, pages 950–957, 1992.

[31] Michael Grant and Stephen Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`, March 2014.

[32] E. van den Berg and M. P. Friedlander. SPGL1: A solver for large-scale sparse reconstruction, June 2007. http://www.cs.ubc.ca/labs/scl/spgl1.

[33] Andreas Themelis and Panagiotis Patrinos. Supermann: a superlinearly convergent algorithm for finding fixed points of nonexpansive operators. *IEEE Transactions on Automatic Control*, 2019.

[34] 20 newsgroups. `http://qwone.com/~jason/20Newsgroups/`.

[35] Miguel Angel Goberna and Marco López-Cerdá. *Linear semi-infinite optimization*. 01 1998.

[36] Walter Rudin. *Principles of Mathematical Analysis*. McGraw-Hill, New York, 1964.

[37] Jerzy K Baksalary and Oskar Maria Baksalary. Particular formulae for the moore–penrose inverse of a columnwise partitioned matrix. *Linear algebra and its applications*, 421(1):16–23, 2007.

[38] Yehoram Gordon. On milman's inequality and random subspaces which escape through a mesh in rn. In *Geometric Aspects of Functional Analysis*, pages 84–106. Springer, 1988.

[39] Michel Ledoux and Michel Talagrand. *Probability in Banach Spaces: isoperimetry and processes*. Springer Science & Business Media, 2013.

[40] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

A   Supplementary Material

In this section, we present proofs of the main results, details on the algorithms and numerical results, and extra figures. We refer to the equations in the main paper as [Main Paper,(#)] to prevent ambiguities.

A.1   Additional figures

Here, we present additional figures (see Figure 11) that support our claims in the main paper. In 1D ReLU NN experiments, we note that even though the weights and biases might take quite different values for each neuron, their activation points , i.e., $-b_i/u_i$ for each neuron i, correspond to the data samples. The distribution of the weights and biases are shown in Figure 1b in this document. In Figure 11b, we also illustrate another rectified ellipsoid set in three dimensions.
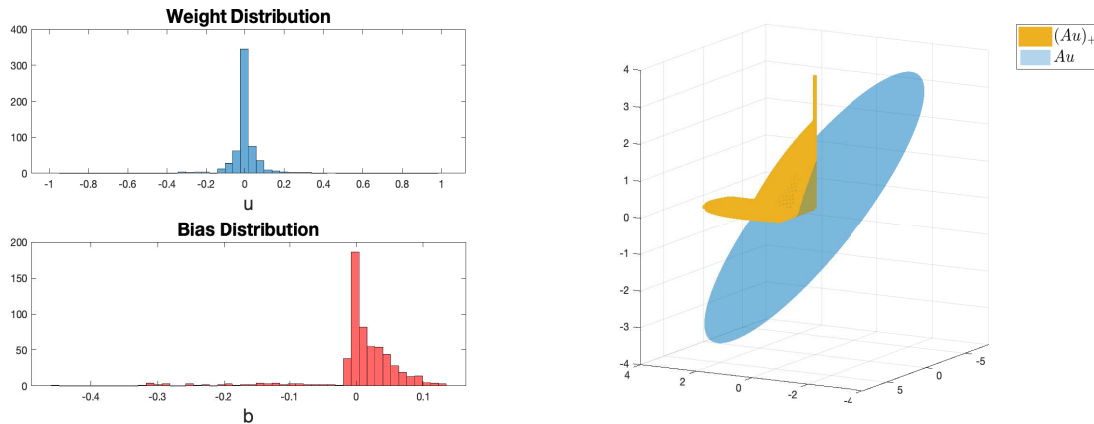
(a) Weight and bias distributions for Figure 1 in the main paper.

(b) 3D Illustration of the rectified ellipsoid.

Figure 11: Additional figures.

A.2   Additional details on the numerical experiments

In this section, we provide further information about our experimental setup.

In the main paper, we evaluate the performance of the introduced approach on several real data sets. For comparison, we also include the performance of a two-layer NN trained with the back-propagation algorithm and the well-known linear least squares approach. For all the experiments, we use the regularization term (also known as weight decay) to let the algorithms generalize well on unseen data [30]. In addition to this, we use the cutting plane based algorithm along with the neurons in [Main Paper,(10)] for our convex approach. In order to solve the convex optimization problems in our approach, we use CVX [31]. However, notice that when dealing with large data sets, e.g., CIFAR-10, plain CVX solvers might need significant amount of memory. In order to circumvent these issues, we use SPGL1 [32] and SuperSCS [33] for large data sets. We also remark that all the data sets we use are publicly available and further information, e.g., training and test sizes, can be obtained through the provided references [25, 26, 27, 34]. Furthermore, we use the same number of hidden neurons for both our approach and the conventional backpropagation based approach to have a fair comparison.

In order to gain further understanding of the connection between implicit regularization and initial standard deviation of the neuron weights, we perform an experiment that is presented in the main paper, i.e., Figure 1. In this experiment, using the backpropagation algorithm, we train two-layers NNs with different initial standard deviations such that each network completely fits the training data. Then, we find the maximum absolute difference between the function fit by the NNs and the ground truth linear interpolation. After averaging our results over many random trials, we obtain Figure 1. The same settings are also used for the experiment using hinge loss.

A.3   Cutting plane algorithm with a bias term

Here, we include the cutting plane algorithm which accommodates a bias term. This is slightly more involved than the case with no bias because of extra constraints. We have the corresponding dual problem as in Theorem 3.1

$$\max_{\mathbf{v}:\mathbf{1}^T\mathbf{v}=0} \mathbf{v}^T\mathbf{y} \text{ s.t. } \left|\mathbf{v}^T\big(\mathbf{A}\mathbf{u}+b\mathbf{1}\big)_+\right| \leq 1 \,, \forall\mathbf{u}\in\mathcal{B}_2, \forall b\in\mathbb{R} \tag{21}$$

and an optimal $(\mathbf{U}^*, \mathbf{b}^*)$ satisfies

$$\|(\mathbf{A}\mathbf{U}^* + \mathbf{1}\mathbf{b}^{*T})_+^T\mathbf{v}^*\|_\infty = 1 \,,$$

where $\mathbf{v}^*$ is the optimal dual variable.

Among infinitely many possible unit norm weights, we need to find the weights that violate the inequality constraint in the dual form, which can be done by solving the following optimization problems

$$\mathbf{u}_1^* = \underset{\mathbf{u},b}{\operatorname{argmax}}\, \mathbf{v}^T(\mathbf{A}\mathbf{u}+b\mathbf{1})_+ \text{ s.t. } \|\mathbf{u}\|_2 \leq 1$$
$$\mathbf{u}_2^* = \underset{\mathbf{u},b}{\operatorname{argmin}}\, \mathbf{v}^T(\mathbf{A}\mathbf{u}+b\mathbf{1})_+ \text{ s.t. } \|\mathbf{u}\|_2 \leq 1.$$

However, the above problem is not convex since ReLU is a convex function. In this case, we can further relax the problem by applying the spike-free relaxation as follows

$$(\hat{\mathbf{u}}_1, \hat{b}_1) = \underset{\mathbf{u},b}{\operatorname{argmax}}\, \mathbf{v}^T\mathbf{A}\mathbf{u}+b\mathbf{v}^T\mathbf{1} \text{ s.t. } \mathbf{A}\mathbf{u}+b\mathbf{1}\succcurlyeq\mathbf{0}, \|\mathbf{u}\|_2 \leq 1$$
$$(\hat{\mathbf{u}}_2, \hat{b}_2) = \underset{\mathbf{u},b}{\operatorname{argmin}}\, \mathbf{v}^T\mathbf{A}\mathbf{u}+b\mathbf{v}^T\mathbf{1} \text{ s.t. } \mathbf{A}\mathbf{u}+b\mathbf{1}\succcurlyeq\mathbf{0}, \|\mathbf{u}\|_2 \leq 1,$$

where we relax the set $\{(\mathbf{A}\mathbf{u}+b\mathbf{1})_+|\mathbf{u}\in\mathbb{R}^d, \|\mathbf{u}\|_2\leq 1\}$ as $\{\mathbf{A}\mathbf{u}+b\mathbf{1}|\mathbf{u}\in\mathbb{R}^d, \|\mathbf{u}\|_2\leq 1\}\cap\mathbb{R}_+^n$. Now, we can find the weights and biases for the hidden layer using convex optimization. However, notice that depending on the sign of $\mathbf{1}^T\mathbf{v}$ one of the problems will be unbounded. Thus, if $\mathbf{1}^T\mathbf{v}\neq 0$, then we can always find a violating constraint, which will make the problem infeasible. However, note that we do not include a bias term for the output layer. If we include the output bias term, then $\mathbf{1}^T\mathbf{v}=0$ will be implicitly enforced via the dual problem.

Based on our analysis, we propose the following convex optimization approach to train the two-layer NN. We first find a violating neuron. After adding these parameters to $\mathbf{U}$ as a column and to $\mathbf{b}$ as a row, we try to solve the original problem. If we cannot find a new violating neuron then we terminate the algorithm. Otherwise, we find the dual parameter for the updated $\mathbf{U}$. We repeat this procedure until the optimality conditions are satisfied (see Algorithm 3 for the pseudocode). Since the constraint is bounded below and $\hat{\mathbf{u}}_j$'s are bounded, Algorithm 3 is guaranteed to converge in finitely many iterations Theorem 11.2 of [35].

Algorithm 3 Cutting Plane based Training Algorithm for Two-Layer NNs (with bias)

1: Initialize $\mathbf{v}$ such that $\mathbf{1}^T\mathbf{v} = 0$
2: **while** there exists a violating neuron **do**
3:      Find $\hat{\mathbf{u}}_1$, $\hat{\mathbf{u}}_2$, $\hat{b}_1$ and $\hat{b}_2$
4:      $\mathbf{U} \leftarrow [\mathbf{U} \ \hat{\mathbf{u}}_1 \ \hat{\mathbf{u}}_2]$
5:      $\mathbf{b} \leftarrow [\mathbf{b}^T \ \hat{b}_1 \ \hat{b}_2]^T$
6:      Find $\mathbf{v}$ using the dual problem
7:      Check the existence of a violating neuron
8: **end while**
9: Solve the problem using $\mathbf{U}$ and $\mathbf{b}$
10: Return $\theta = (\mathbf{U}, \mathbf{b}, \mathbf{w})$

A.4   Infinite size neural networks

Here we briefly review infinite size, i.e., infinite width, two-layer NNs [7]. We refer the reader to [5, 6] for further background and connections to our work. Consider an arbitrary measurable input space $\mathcal{X}$ with a set of continuous basis functions $\phi_{\mathbf{u}} : \mathcal{X} \to \mathbb{R}$ parameterized by $\mathbf{u} \in \mathcal{B}_2$. We then consider real-valued Radon measures equipped with the uniform norm [36]. For a signed Radon measure $\boldsymbol{\mu}$, we define the infinite size NN output for the input $\mathbf{x} \in \mathcal{X}$ as

$$f(\mathbf{x}) = \int_{\mathbf{u} \in \mathcal{B}_2} \phi_{\mathbf{u}}(\mathbf{x}) d\boldsymbol{\mu}(\mathbf{u}) \,.$$

The total variation norm of the signed measure $\boldsymbol{\mu}$ is defined as the supremum of $\int_{\mathbf{u} \in \mathcal{B}_2} q(\mathbf{u}) d\boldsymbol{\mu}(\mathbf{u})$ over all continuous functions $q(\mathbf{u})$ that satisfy $|q(\mathbf{u})| \leq 1$. Now we consider the ReLU basis functions $\phi_{\mathbf{u}}(\mathbf{x}) = (\mathbf{x}^T \mathbf{u})_+$. For finitely many neurons, the network output is given by

$$f(\mathbf{x}) = \sum_{j=1}^{m} \phi_{\mathbf{u}_j}(\mathbf{x}) w_j \,,$$

which corresponds to the signed measure $\boldsymbol{\mu} = \sum_{j=1}^{m} w_j \delta(\mathbf{u} - \mathbf{u}_j)$, where δ is the Dirac delta function. And the total variation norm $\|\boldsymbol{\mu}\|_{TV}$ of $\boldsymbol{\mu}$ reduces to the ℓ_1 norm $\|\mathbf{w}\|_1$.

     The infinite dimensional version of the problem [Main Paper, (4)] corresponds to

$$\min \|\boldsymbol{\mu}\|_{TV}$$
$$\text{s.t.} \quad f(\mathbf{x}_i) = y_i \,, \forall i \in [n] \,.$$

For finitely many neurons, i.e., when the measure $\boldsymbol{\mu}$ is a mixture of Dirac delta basis functions, the equivalent problem is

$$\min \|\mathbf{w}\|_1$$
$$\text{s.t.} \quad f(\mathbf{x}_i) = y_i \,, \forall i \in [n] \,.$$

which is identical to [Main Paper, (4)]. Similar results also hold with regularized objective functions, different loss functions and vector outputs.

A.5   Proofs of the main results

In this section, we present the proofs of the theorems and lemmas provided in the main paper.

Proof of Lemma 2.1. For any $\theta \in \Theta$, we can rescale the parameters as $\bar{\mathbf{u}}_j = \alpha_j \mathbf{u}_j$, $\bar{b}_j = \alpha_j b_j$ and $\bar{w}_j = w_j/\alpha_j$, for any $\alpha_j > 0$. Then, [Main Paper, (1)] becomes

$$f_{\bar{\theta}}(\mathbf{A}) = \sum_{j=1}^{m} \bar{w}_j (\mathbf{A}\bar{\mathbf{u}}_j + \bar{b}_j \mathbf{1})_+ = \sum_{j=1}^{m} \frac{w_j}{\alpha_j}(\alpha_j \mathbf{A}\mathbf{u}_j + \alpha_j b_j \mathbf{1})_+ = \sum_{j=1}^{m} w_j (\mathbf{A}\mathbf{u}_j + b_j \mathbf{1})_+,$$

which proves $f_{\theta}(\mathbf{A}) = f_{\bar{\theta}}(\mathbf{A})$. In addition to this, we have the following basic inequality

$$\frac{1}{2} \sum_{j=1}^{m} (w_j^2 + \|\mathbf{u}_j\|_2^2) \geq \sum_{j=1}^{m} (|w_j|\, \|\mathbf{u}_j\|_2),$$

where the equality is achieved with the scaling choice $\alpha_j = \left(\frac{|w_j|}{\|\mathbf{u}_j\|_2}\right)^{\frac{1}{2}}$. Since the scaling operation does not change the right-hand side of the inequality, we can set $\|\mathbf{u}_j\|_2 = 1, \forall j$. Therefore, the right-hand side becomes $\|\mathbf{w}\|_1$. $\qquad\square$

Proof of Lemma 2.2. Consider the following problem

$$\min_{\theta \in \Theta} \|\mathbf{w}\|_1 \text{ s.t. } f_{\theta}(\mathbf{A}) = \mathbf{y}, \|\mathbf{u}_j\|_2 \leq 1, \forall j,$$

where the unit norm equality constraint is relaxed. Let us assume that for a certain index j, we obtain $\|\mathbf{u}_j\|_2 < 1$ with $w_j \neq 0$ as the optimal solution of the above problem. This shows that the unit norm inequality constraint is not active for $\mathbf{u}_j$, and hence removing the constraint for $\mathbf{u}_j$ will not change the optimal solution. However, when we remove the constraint, $\|\mathbf{u}_j\|_2 \to \infty$ reduces the objective value since it yields $w_j = 0$. Hence, we have a contradiction, which proves that all the constraints that correspond to a nonzero w_j must be active for an optimal solution. $\qquad\square$

Proof of Lemma 2.3. The first condition immediately implies that $\{(\mathbf{A}\mathbf{u})_+ | \mathbf{u} \in \mathcal{B}_2\} \subseteq \mathbf{A}\mathcal{B}_2$. Since we also have $\{(\mathbf{A}\mathbf{u})_+ | \mathbf{u} \in \mathcal{B}_2\} \subseteq \mathbb{R}_+^n$, it holds that $\{(\mathbf{A}\mathbf{u})_+ | \mathbf{u} \in \mathcal{B}_2\} \subseteq \mathbf{A}\mathcal{B}_2 \cap \mathbb{R}_+^n$. The projection of $\mathbf{A}\mathcal{B}_2 \cap \mathbb{R}_+^n$ onto the positive orthant is a subset of $\mathcal{Q}_{\mathbf{A}}$, and consequently we have $\mathcal{Q}_{\mathbf{A}} = \mathbf{A}\mathcal{B}_2 \cap \mathbb{R}_+^n$.
The second conditions follow from the min-max representation

$$\max_{\mathbf{u} \in \mathcal{B}_2} \min_{\mathbf{z}:\, \mathbf{A}\mathbf{z} = (\mathbf{A}\mathbf{u})_+} \|\mathbf{z}\|_2 \leq 1 \iff [MainPaper, (6)],$$

by noting that $(\mathbf{I}_n - \mathbf{A}\mathbf{A}^\dagger)(\mathbf{A}\mathbf{u})_+ = \mathbf{0}$ if and only if there exists $\mathbf{z}$ such that $\mathbf{A}\mathbf{z} = (\mathbf{A}\mathbf{u})_+$, which in that case provided by $\mathbf{A}^\dagger(\mathbf{A}\mathbf{u})_+$. The third condition follows from the fact that the minimum norm solution to $\mathbf{A}\mathbf{z} = (\mathbf{A}\mathbf{u})_+$ is given by $\mathbf{A}^\dagger(\mathbf{A}\mathbf{u})_+$ under the full row rank assumption on $\mathbf{A}$, which in turn implies $\mathbf{I}_n - \mathbf{A}\mathbf{A}^\dagger = \mathbf{0}$. $\qquad\square$

Proof of Lemma 2.4. We have

$$\max_{\mathbf{u}:\, \|\mathbf{u}\|_2 \leq 1} \|\mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}(\mathbf{A}\mathbf{u})_+\|_2 \leq \sigma_{\max}(\mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}) \max_{\mathbf{u}:\, \|\mathbf{u}\|_2 \leq 1} \|(\mathbf{A}\mathbf{u})_+\|_2$$

$$= \sigma_{\min}^{-1}(\mathbf{A}) \max_{\mathbf{u}:\, \|\mathbf{u}\|_2 \leq 1} \|(\mathbf{A}\mathbf{u})_+\|_2$$

$$\leq \sigma_{\min}^{-1}(\mathbf{A}) \max_{\mathbf{u}:\, \|\mathbf{u}\|_2 \leq 1} \|\mathbf{A}\mathbf{u}\|_2$$

$$\leq \sigma_{\min}^{-1}(\mathbf{A}) \sigma_{\max}(\mathbf{A})$$

$$\leq 1.$$

where the last inequality follows from the fact that $\mathbf{A}$ is whitened. $\qquad\square$

Proof of Lemma 2.5. Let us consider a data matrix $\mathbf{A}$ such that $\mathbf{A} = \mathbf{c}\mathbf{a}^T$, where $\mathbf{c} \in \mathbb{R}_+^n$ and $\mathbf{a} \in \mathbb{R}^d$. Then, $(\mathbf{A}\mathbf{u})_+ = \mathbf{c}(\mathbf{a}^T\mathbf{u})_+$. If $(\mathbf{a}^T\mathbf{u})_+ = 0$, then we can select $\mathbf{z} = \mathbf{0}$ to satisfy the spike-free condition $(\mathbf{c}\mathbf{a}^T\mathbf{u})_+ = \mathbf{A}\mathbf{z}$. If $(\mathbf{a}^T\mathbf{u})_+ \neq 0$, then $(\mathbf{A}\mathbf{u})_+ = \mathbf{c}\mathbf{a}^T\mathbf{u} = \mathbf{A}\mathbf{u}$, where the spike-free condition can be trivially satisfied with the choice of $\mathbf{z} = \mathbf{u}$. $\square$

Proof of Lemma 2.6. The extreme point along the direction of $\mathbf{v}$ can be found as follows

$$\operatorname*{argmax}_{u,b} \sum_{i=1}^n v_i(a_i u + b)_+ \text{ s.t. } |u| = 1, \tag{22}$$

Since each neuron separates the samples into two sets, for some samples, ReLU will be active, i.e., $\mathcal{S} = \{i | a_i u + b \geq 0\}$, and for the others, it will be inactive, i.e., $\mathcal{S}^c = \{j | a_j u + b < 0\} = [n]/\mathcal{S}$. Thus, we modify (22) as

$$\operatorname*{argmax}_{u,b} \sum_{i \in \mathcal{S}} v_i(a_i u + b) \text{ s.t. } a_i u + b \geq 0, \forall i \in \mathcal{S}, a_j u + b \leq 0, \forall j \in \mathcal{S}^c, |u| = 1. \tag{23}$$

In (23), u can only take two values, i.e., ± 1. Thus, we can separately solve the optimization problem for each case and then take the maximum one as the optimal. Let us assume that $u = 1$. Then, (23) reduces to finding the optimal bias. We note that due to the constraints in (23), $-a_i \leq b \leq -a_j, \forall i \in \mathcal{S}, \forall j \in \mathcal{S}^c$. Thus, the range for the possible bias values is $[\max_{i \in \mathcal{S}}(-a_i), \ \min_{j \in \mathcal{S}^c}(-a_j)]$. Therefore, depending on the direction $\mathbf{v}$, the optimal bias can be selected as follows

$$b_v = \begin{cases} \max_{i \in \mathcal{S}}(-a_i), & \text{if } \sum_{i \in \mathcal{S}} v_i \leq 0 \\ \min_{j \in \mathcal{S}^c}(-a_j), & \text{otherwise} \end{cases}. \tag{24}$$

Similar arguments also hold for $u = -1$ and the argmin version of (22). Note that when $\sum_{i \in \mathcal{S}} v_i = 0$, the value of the bias does not change the objective in (23). Thus, all the bias values in the range $[\max_{i \in \mathcal{S}}(-a_i), \ \min_{j \in \mathcal{S}^c}(-a_j)]$ become optimal. In such cases, there might exists multiple optimal solutions for the training problem. $\square$

Proof of Lemma 2.7. For the extreme point in the span of $\mathbf{e}_i$, we need to solve the following optimization problem

$$\operatorname*{argmax}_{\mathbf{u},b} (\mathbf{a}_i^T \mathbf{u} + b) \text{ s.t. } \mathbf{a}_j^T \mathbf{u} + b \leq 0, \forall i \neq j, \|\mathbf{u}\|_2 = 1. \tag{25}$$

Then the Lagrangian of (25) is

$$L(\boldsymbol{\lambda}, \mathbf{u}, b) = \mathbf{a}_i^T \mathbf{u} + b - \sum_{\substack{j=1 \\ j \neq i}}^n \lambda_j (\mathbf{a}_j^T \mathbf{u} + b), \tag{26}$$

where we do not include the unit norm constraint for $\mathbf{u}$. For (26), $\boldsymbol{\lambda}$ must satisfy $\boldsymbol{\lambda} \succcurlyeq \mathbf{0}$ and $\mathbf{1}^T \boldsymbol{\lambda} = 1$. With these specifications, the problem can be written as

$$\min_{\boldsymbol{\lambda}} \max_{\mathbf{u}} \mathbf{u}^T \left(\mathbf{a}_i - \sum_{\substack{j=1 \\ j \neq i}}^n \lambda_j \mathbf{a}_j \right) \text{ s.t. } \boldsymbol{\lambda} \succcurlyeq \mathbf{0}, \mathbf{1}^T \boldsymbol{\lambda} = 1, \|\mathbf{u}\|_2 = 1. \tag{27}$$

Since the $\mathbf{u}$ vector that maximizes (27) is the normalized version of the term inside the parenthesis above, the problem reduces to

$$\min_{\boldsymbol{\lambda}} \left\| \mathbf{a}_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} \lambda_j \mathbf{a}_j \right\|_2 \text{ s.t. } \boldsymbol{\lambda} \succcurlyeq \mathbf{0}, \mathbf{1}^T \boldsymbol{\lambda} = 1. \tag{28}$$

After solving the convex problem (28) for each i, we can find the corresponding neurons as follows

$$\mathbf{u}_i = \frac{\mathbf{a}_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} \lambda_j \mathbf{a}_j}{\left\| \mathbf{a}_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} \lambda_j \mathbf{a}_j \right\|_2} \text{ and } b_i = \min_{j \neq i}(-\mathbf{a}_j^T \mathbf{u}_i),$$

where the bias computation follows from the constraint in (25). $\qquad \square$

Proof of Lemma 2.8. For any $\boldsymbol{\alpha} \in \mathbb{R}^n$, the extreme point along the direction of $\boldsymbol{\alpha}$ can be found by solving the following optimization problem

$$\operatorname*{argmax}_{\mathbf{u},b} \boldsymbol{\alpha}^T (\mathbf{A}\mathbf{u} + b\mathbf{1})_+ \text{ s.t. } \|\mathbf{u}\|_2 \leq 1 \tag{29}$$

where the optimal $(\mathbf{u}, b)$ groups samples into two sets so that some of them activates ReLU with the indices $\mathcal{S} = \{i | \mathbf{a}_i^T \mathbf{u} + b \geq 0\}$ and the others deactivate it with the indices $\mathcal{S}^c = \{j | \mathbf{a}_j^T \mathbf{u} + b < 0\} = [n]/\mathcal{S}$. Using this, we equivalently write (29) as

$$\max_{\mathbf{u},b} \sum_{i \in \mathcal{S}} \alpha_i (\mathbf{a}_i^T \mathbf{u} + b) \text{ s.t. } (\mathbf{a}_i^T \mathbf{u} + b) \geq 0, \forall i \in \mathcal{S}, (\mathbf{a}_j^T \mathbf{u} + b) \leq 0, \forall j \in \mathcal{S}^c, \|\mathbf{u}\|_2 \leq 1,$$

which has the following dual form

$$\min_{\boldsymbol{\lambda}, \boldsymbol{\nu}} \max_{\mathbf{u},b} \mathbf{u}^T \left(\sum_{i \in \mathcal{S}} (\alpha_i + \lambda_i) \mathbf{a}_i - \sum_{j \in \mathcal{S}^c} \nu_j \mathbf{a}_j \right) \text{ s.t. } \boldsymbol{\lambda}, \boldsymbol{\nu} \succcurlyeq \mathbf{0}, \sum_{i \in \mathcal{S}} (\alpha_i + \lambda_i) = \sum_{j \in \mathcal{S}^c} \nu_j, \|\mathbf{u}\|_2 \leq 1.$$

Thus, we obtain the following neuron and bias choice for the extreme point

$$\mathbf{u}_{\alpha} = \frac{\sum_{i \in \mathcal{S}} (\alpha_i + \lambda_i) \mathbf{a}_i - \sum_{j \in \mathcal{S}^c} \nu_j \mathbf{a}_j}{\| \sum_{i \in \mathcal{S}} (\alpha_i + \lambda_i) \mathbf{a}_i - \sum_{j \in \mathcal{S}^c} \nu_j \mathbf{a}_j \|_2} \text{ and } b_{\alpha} = \begin{cases} \max_{i \in \mathcal{S}}(-\mathbf{a}_i^T \mathbf{u}), & \text{if } \sum_{i \in \mathcal{S}} \alpha_i \leq 0 \\ \min_{j \in \mathcal{S}^c}(-\mathbf{a}_j^T \mathbf{u}), & \text{otherwise} \end{cases}.$$

$\qquad \square$

Proof of Theorem 3.1 and Corollary 3.1. We first note that the dual of [Main Paper,(4)] with respect to $\mathbf{w}$ is

$$\min_{\theta \in \Theta \setminus \{\mathbf{w}\}} \max_{\mathbf{v}} \mathbf{v}^T \mathbf{y} \text{ s.t. } \|(\mathbf{A}\mathbf{U})_+^T \mathbf{v}\|_\infty \leq 1, \ \|\mathbf{u}_j\|_2 \leq 1, \forall j.$$

Then, we can reformulate the problem as follows

$$P^* = \min_{\theta \in \Theta \setminus \{\mathbf{w}\}} \max_{\mathbf{v}} \mathbf{v}^T \mathbf{y} + \mathcal{I}(\|(\mathbf{A}\mathbf{U})_+^T \mathbf{v}\|_\infty \leq 1), \text{ s.t. } \|\mathbf{u}_j\|_2 \leq 1, \forall j.$$

28

where $\mathcal{I}(\|(\mathbf{AU})_+^T\mathbf{v}\|_\infty \leq 1)$ is the characteristic function of the set $\|(\mathbf{AU})_+^T\mathbf{v}\|_\infty \leq 1$, which is defined as

$$\mathcal{I}(\|(\mathbf{AU})_+^T\mathbf{v}\|_\infty \leq 1) = \begin{cases} 0 & \text{if } \|(\mathbf{AU})_+^T\mathbf{v}\|_\infty \leq 1 \\ -\infty & \text{otherwise} \end{cases}.$$

Since the set $\|(\mathbf{AU})_+^T\mathbf{v}\|_\infty \leq 1$ is closed, the function $\Phi(\mathbf{v},\mathbf{U}) = \mathbf{v}^T\mathbf{y} + \mathcal{I}(\|(\mathbf{AU})_+^T\mathbf{v}\|_\infty \leq 1)$ is the sum of a linear function and an upper-semicontinuous indicator function and therefore upper-semicontinuous. The constraint on $\mathbf{U}$ is convex and compact. We use P^* to denote the value of the above min-max program. Exchanging the order of min and max we obtain the dual problem given in [Main Paper,(12)], which establishes a lower bound D^* for the above problem:

$$P^* \geq D^* = \max_{\mathbf{v}} \min_{\theta \in \Theta \setminus \{\mathbf{w}\}} \mathbf{v}^T\mathbf{y} + \mathcal{I}(\|(\mathbf{AU})_+^T\mathbf{v}\|_\infty \leq 1), \text{ s.t. } \|\mathbf{u}_j\|_2 \leq 1, \forall j,$$

$$= \max_{\mathbf{v}} \mathbf{v}^T\mathbf{y}, \text{ s.t. } \|(\mathbf{AU})_+^T\mathbf{v}\|_\infty \leq 1 \ \forall \mathbf{u}_j : \|\mathbf{u}_j\|_2 \leq 1, \forall j,$$

$$= \max_{\mathbf{v}} \mathbf{v}^T\mathbf{y}, \text{ s.t. } \|(\mathbf{Au})_+^T\mathbf{v}\|_\infty \leq 1 \ \forall \mathbf{u} : \|\mathbf{u}\|_2 \leq 1,$$

We now show that strong duality holds for infinite size NNs. The dual of the semi-infinite program in [Main Paper, (12)] is given by (see Section 2.2 of [35] and also [7])

$$\min \|\boldsymbol{\mu}\|_{TV}$$

$$\text{s.t.} \int_{\mathbf{u} \in \mathcal{B}_2} (\mathbf{Au})_+ d\boldsymbol{\mu}(\mathbf{u}) = \mathbf{y},$$

where TV is the total variation norm of the Radon measure $\boldsymbol{\mu}$. This expression coincides with the infinite-size NN as given in Section A.4, and therefore strong duality holds. Next we invoke the semi-infinite optimality conditions for the dual problem in [Main Paper,(12)], in particular we apply Theorem 7.2 of [35]. We first define the set

$$\mathbf{K} = \mathbf{cone}\left\{\begin{pmatrix} s(\mathbf{Au})_+ \\ 1 \end{pmatrix}, \mathbf{u} \in \mathcal{B}_2, s \in \{-1, +1\}; \begin{pmatrix} \mathbf{0} \\ -1 \end{pmatrix}\right\}.$$

Note that $\mathbf{K}$ is the union of finitely many convex closed sets, since the function $(\mathbf{Au})_+$ can be expressed as the union of finitely many convex closed sets. Therefore the set $\mathbf{K}$ is closed. By Theorem 5.3 [35], this implies that the set of constraints in [Main Paper,(12)] forms a Farkas-Minkowski system. By Theorem 8.4 of [35], primal and dual values are equal, given that the system is consistent. Moreover, the system is discretizable, i.e., there exists a sequence of problems with finitely many constraints whose optimal values approach to the optimal value of [Main Paper,(12)]. The optimality conditions in Theorem 7.2 [35] implies that $\mathbf{y} = (\mathbf{AU}^*)_+\mathbf{w}^*$ for some vector $\mathbf{w}^*$. Since the primal and dual values are equal, we have $\mathbf{v}^{*T}\mathbf{y} = \mathbf{v}^{*T}(\mathbf{AU}^*)_+\mathbf{w}^* = \|\mathbf{w}^*\|_1$, which shows that the primal-dual pair $(\{\mathbf{w}^*,\mathbf{U}^*\},\mathbf{v}^*)$ is optimal. Thus, the optimal neuron weights $\mathbf{U}^*$ satisfy $\|(\mathbf{AU}^*)_+^T\mathbf{v}^*\|_\infty = 1$. $\qquad\square$

Proof of Theorem 3.2. We first assume that zero training error can be achieved with m_1 neurons. Then, we obtain the dual of [Main Paper,(4)] with $m = m_1$

$$P_f^* = \min_{\theta \in \Theta \setminus \{\mathbf{w},m\}} \max_{\mathbf{v}} \mathbf{v}^T\mathbf{y} \text{ s.t. } \|(\mathbf{AU})_+^T\mathbf{v}\|_\infty \leq 1, \|\mathbf{u}_j\|_2 \leq 1, \forall j \in [m_1]. \tag{30}$$

Exchanging the order of min and max establishes a lower bound for (30)

$$P_f^* \geq D_f^* = \max_{\mathbf{v}} \min_{\theta \in \Theta \setminus \{\mathbf{w}, m\}} \mathbf{v}^T \mathbf{y} + \mathcal{I}(\|(\mathbf{AU})_+^T \mathbf{v}\|_\infty \leq 1), \text{ s.t. } \|\mathbf{u}_j\|_2 \leq 1, \forall j \in [m_1]. \qquad (31)$$

If we denote the optimal parameters to (30) as $\mathbf{U}^*$ and $\mathbf{v}^*$, then $|(\mathbf{AU}^*)_+^T \mathbf{v}^*| = \mathbf{1}$ must hold, i.e., all the optimal neuron weights must achieve the extreme point of the inequality constraint. To prove this, let us consider an optimal neuron $\mathbf{u}_j^*$, which has a nonzero weight $w_j \neq 0$ and $|(\mathbf{Au}_j^*)_+^T \mathbf{v}^*| < 1$. Then, even if we remove the inequality constraint for $\mathbf{u}_j^*$ in (30), the optimal objective value will not change. However, if we remove it, then $\mathbf{u}_j^*$ will no longer contribute to $(\mathbf{AU})_+ \mathbf{w} = \mathbf{y}$. Then, we can achieve a smaller objective value, i.e., $\|\mathbf{w}\|_1$, by simply setting $w_j = 0$. Thus, we obtain a contradiction, which proves that the inequality constraints that correspond to the neurons with nonzero weight, $w_j \neq 0$, must achieve the extreme point for the optimal solution, i.e., $|(\mathbf{Au}_j^*)_+^T \mathbf{v}^*| = 1, \forall j \in [m]$.

Based on this observation, we have

$$
\begin{aligned}
P_f^* = \min_{\theta \in \Theta \setminus \{\mathbf{w}, m\}} \max_{\mathbf{v}} \mathbf{v}^T \mathbf{y} \qquad &\geq \qquad \min_{\theta \in \Theta \setminus \{\mathbf{w}\}} \max_{\mathbf{v}} \mathbf{v}^T \mathbf{y} \\
\text{s.t. } \|(\mathbf{AU})_+^T \mathbf{v}\|_\infty \leq 1, \ \|\mathbf{u}_j\|_2 \leq 1, \forall j \in [m_1] \quad &\quad \text{s.t. } \|(\mathbf{AU})_+^T \mathbf{v}\|_\infty \leq 1, \ \|\mathbf{u}_j\|_2 \leq 1, \forall j \\
&\geq \qquad \max_{\mathbf{v}} \min_{\theta \in \Theta \setminus \{\mathbf{w}\}} \mathbf{v}^T \mathbf{y} \\
&\quad \text{s.t. } \|(\mathbf{AU})_+^T \mathbf{v}\|_\infty \leq 1, \ \|\mathbf{u}_j\|_2 \leq 1, \forall j \\
&= \qquad \max_{\mathbf{v}} \min_{\theta \in \Theta \setminus \{\mathbf{w}, m\}} \mathbf{v}^T \mathbf{y} \\
&\quad \text{s.t. } (\|(\mathbf{AU})_+^T \mathbf{v}\|_\infty \leq 1, \ \|\mathbf{u}_j\|_2 \leq 1, \forall j \in [m_1] \\
&= \qquad D_f^* = D^* \qquad (32)
\end{aligned}
$$

where the first inequality is based on the fact that an infinite width NN can always find a solution with the objective value lower than or equal to the objective value of a finite width NN. The second inequality follows from (31). More importantly, the equality in the third line follows from our observation above, i.e., neurons that are not the extreme point of the inequality in (30) do not change the objective value. Therefore, by (32), we prove that weak duality holds for a finite width NN, i.e., $P_f^* \geq P^* \geq D_f^* = D^*$. $\qquad\square$

Proof of Theorem 3.3. We first assume that there exist finitely many, say m_e, extreme points. Thus, we can construct a weight matrix $\mathbf{U}_e \in \mathbb{R}^{d \times m_e}$ that consists of all the possible extreme points. Then, the dual of [Main Paper,(4)] with $\mathbf{U} = \mathbf{U}_e$

$$D_f^* = \max_{\mathbf{v}} \mathbf{v}^T \mathbf{y} \text{ s.t. } \|(\mathbf{AU}_e)_+^T \mathbf{v}\|_\infty \leq 1, \qquad (33)$$

Then, we have

$$
\begin{aligned}
P^* = \min_{\theta \in \Theta \setminus \{\mathbf{w}\}} \max_{\mathbf{v}} \mathbf{v}^T \mathbf{y} \qquad &\geq \qquad \max_{\mathbf{v}} \min_{\theta \in \Theta \setminus \{\mathbf{w}\}} \mathbf{v}^T \mathbf{y} \\
\text{s.t } \|(\mathbf{AU})_+^T \mathbf{v}\|_\infty \leq 1, \ \|\mathbf{u}_j\|_2 \leq 1, \forall j \quad &\quad \text{s.t. } \|(\mathbf{AU})_+^T \mathbf{v}\|_\infty \leq 1, \ \|\mathbf{u}_j\|_2 \leq 1, \forall j \\
&= \qquad \max_{\mathbf{v}} \mathbf{v}^T \mathbf{y} \\
&\quad \text{s.t. } (\|(\mathbf{AU}_e)_+^T \mathbf{v}\|_\infty \leq 1 \\
&= \qquad D_f^* = D^* \qquad (34)
\end{aligned}
$$

where the first inequality follows from changing order of min-max to obtain a lower bound and the equality in the second line follows from Corollary 3.1 and our observation above, i.e., neurons that are not the extreme point of the inequality in (33) do not change the objective value.

From the fact that an infinite width NN can always find a solution with the objective value lower than or equal to the objective value of a finite width NN, we have

$$P_f^* = \min_{\theta \in \Theta \setminus \{\mathbf{U}, m\}} \|\mathbf{w}\|_1 \qquad \geq \qquad P^* = \min_{\theta \in \Theta} \|\mathbf{w}\|_1 \tag{35}$$
$$\text{s.t. } (\mathbf{A}\mathbf{U}_e)_+ \mathbf{w} = \mathbf{y} \qquad \qquad \text{s.t. } (\mathbf{A}\mathbf{U})_+ \mathbf{w} = \mathbf{y}, \ \|\mathbf{u}_j\|_2 \leq 1, \forall j,$$

where P^* is the optimal value of the original problem with infinitely many neurons. Now, notice that the optimization problem on the left hand side of (35) is convex since it is an ℓ_1-norm minimization problem with linear equality constraints. Therefore, strong duality holds for this problem, i.e., $P_f^* = D_f^*$ and we have $P^* \geq D^* = D_f^*$. Using this result along with (34), we prove that strong duality holds for a finite width NN, i.e., $P_f^* = P^* = D^* = D_f^*$. $\qquad\qquad\qquad\square$

Proof of Proposition 3.1. Here, we particularly examine the problem in [Main Paper, (4)] when we have a one dimensional dataset, i.e., $\{a_i, y_i\}_{i=1}^n$. Then, [Main Paper, (4)] can be modified as

$$\min_{\theta \in \Theta} \|\mathbf{w}\|_1 \text{ s.t. } (\mathbf{a}\mathbf{u}^T + \mathbf{1}\mathbf{b}^T)_+ \mathbf{w} = \mathbf{y}, |u_j| \leq 1, \forall j. \tag{36}$$

Then, using Lemma 2.6, we can construct the following matrix

$$\mathbf{A}_e = (\mathbf{a}\mathbf{u}^{*^T} + \mathbf{1}\mathbf{b}^{*^T})_+,$$

where $\mathbf{u}^*$ and $\mathbf{b}^*$ consist of all possible extreme points. Using this definition and Corollary 3.2, we can rewrite (36) as

$$\min_{\mathbf{w}} \|\mathbf{w}\|_1 \text{ s.t. } \mathbf{A}_e \mathbf{w} = \mathbf{y}. \tag{37}$$

In the following, we first derive optimality conditions for (37) and then provide an analytic counter example to disprove uniqueness. Then, we also follow the same steps for the regularized version of (37).

Equality constraint: The optimality conditions for (37) are

$$\begin{aligned} \mathbf{A}_e \mathbf{w}^* &= \mathbf{y} \\ \mathbf{A}_{e,s}^T \mathbf{v}^* + \text{sign}(\mathbf{w}_s^*) &= 0 \\ \|\mathbf{A}_{e,s^c}^T \mathbf{v}^*\|_\infty &\leq 1, \end{aligned} \tag{38}$$

where the subscript s denotes the entries of a vector (or columns for matrices) that correspond to a nonzero weight, i.e. $w_i \neq 0$, and the subscript s^c denotes the remaining entries (or columns). We aim to find an optimal primal-dual pair that satisfies (38).

Now, let us consider a specific dataset, i.e., $\mathbf{a} = [-2 \ -1 \ 0 \ 1 \ 2]^T$ and $\mathbf{y} = [1 \ -1 \ 1 \ 1 \ -1]^T$, and yields the following

$$\mathbf{A}_e = (\mathbf{a}\mathbf{u}^{*^T} + \mathbf{1}\mathbf{b}^{*^T}) = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 2 & 3 & 4 \\ 1 & 0 & 0 & 0 & 0 & 1 & 2 & 3 \\ 2 & 1 & 0 & 0 & 0 & 0 & 1 & 2 \\ 3 & 2 & 1 & 0 & 0 & 0 & 0 & 1 \\ 4 & 3 & 2 & 1 & 0 & 0 & 0 & 0 \end{bmatrix},$$

where $\mathbf{u}^{*T} = [1\ 1\ 1\ 1\ -1\ -1\ -1\ -1]$ and $\mathbf{b}^{*T} = [2\ 1\ 0\ -1\ -1\ 0\ 1\ 2]$. Solving (37) for this dataset gives

$$\mathbf{v}^* = \begin{bmatrix} 1 \\ -3 \\ 2 \\ 1 \\ -1 \end{bmatrix} \text{ and } \mathbf{w}^* = \begin{bmatrix} 0 \\ 6419/5000 \\ -3919/2500 \\ -8581/5000 \\ 13581/5000 \\ -1081/2500 \\ -1419/5000 \\ 0 \end{bmatrix} \implies \|\mathbf{w}^*\|_1 = 8.$$

We can also achieve the same objective value by using the following matrix

$$\hat{\mathbf{A}}_e = (\mathbf{a}\hat{\mathbf{u}}^T + \mathbf{1}\hat{\mathbf{b}}^T) = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 2 & 2.5 & 4 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1.5 & 3 \\ 2 & 1 & 0 & 0 & 0 & 0 & 0.5 & 2 \\ 3 & 2 & 1 & 0.5 & 0 & 0 & 0 & 1 \\ 4 & 3 & 2 & 1.5 & 0 & 0 & 0 & 0 \end{bmatrix},$$

where $\hat{\mathbf{u}}^T = [1\ 1\ 1\ 1\ -1\ -1\ -1\ -1]$ and $\hat{\mathbf{b}}^T = [2\ 1\ 0\ -0.5\ -1\ 0\ 0.5\ 2]$. Solving (37) for this dataset yields

$$\hat{\mathbf{v}} = \begin{bmatrix} 1 \\ -11/4 \\ 5/4 \\ 7/4 \\ -5/4 \end{bmatrix} \text{ and } \hat{\mathbf{w}} = \begin{bmatrix} 0 \\ 4/3 \\ 0 \\ -10/3 \\ 8/3 \\ 0 \\ -2/3 \\ 0 \end{bmatrix} \implies \|\hat{\mathbf{w}}\|_1 = 8.$$

We also note that both solutions satisfy the optimality conditions in (38).

Regularized case: The regularized version of (37) is as follows

$$\min_{\mathbf{w}} \beta\|\mathbf{w}\|_1 + \frac{1}{2n}\|\mathbf{A}_e\mathbf{w} - \mathbf{y}\|_2^2, \tag{39}$$

where the optimal solution $\mathbf{w}^*$ satisfies

$$\frac{1}{n}\mathbf{A}_{e,s}^T(\mathbf{A}_e\mathbf{w}^* - \mathbf{y}) + \beta\mathrm{sign}(\mathbf{w}_s^*) = 0$$
$$\|\mathbf{A}_{e,s^c}^T(\mathbf{A}_e\mathbf{w}^* - \mathbf{y})\|_\infty \leq \beta n, \tag{40}$$

where the subscript s denotes the entries of a vector (or columns for matrices) that correspond to a nonzero weight, i.e. $w_i \neq 0$, and the subscript s^c denotes the remaining entries (or columns). Now, let us consider a specific dataset, i.e., $\mathbf{a} = [-2\ -1\ 0\ 1\ 2]^T$ and $\mathbf{y} = [1\ -1\ 1\ 1\ -1]^T$. We then construct the following matrix

$$\mathbf{A}_e = (\mathbf{a}\mathbf{u}^{*T} + \mathbf{1}\mathbf{b}^{*T}) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 2 & 2.5 & 3 & 4 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1.5 & 2 & 3 \\ 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0.5 & 1 & 2 \\ 3 & 2 & 1 & 0.5 & 0 & 0 & 0 & 0 & 0 & 1 \\ 4 & 3 & 2 & 1.5 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$
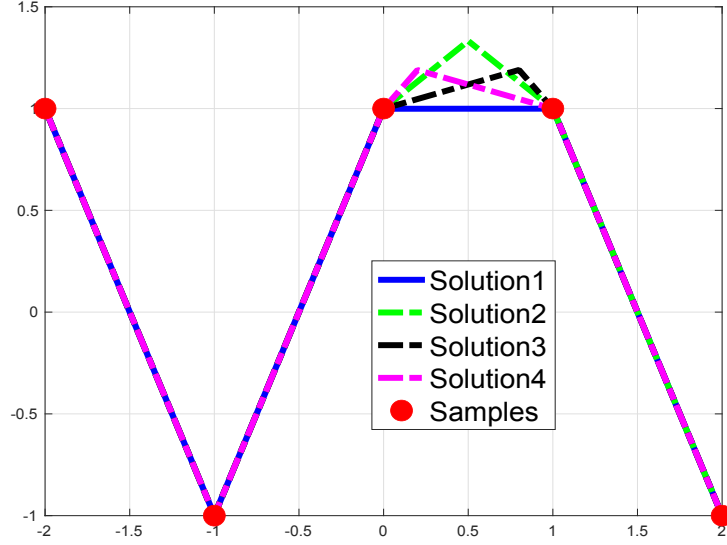
Figure 12: Solutions provided by $\mathbf{w}_1$ and $\mathbf{w}_2$ for the problem in (39).

where $\mathbf{u}^{*T} = [1\ 1\ 1\ 1\ 1\ -1\ -1\ -1\ -1\ -1]$ and $\mathbf{b}^{*T} = [2\ 1\ 0\ -0.5\ -1\ -1\ 0\ 0.5\ 1\ 2]$. For this dataset with $\beta = 10^{-4}$, the optimal value of (39) can be achieved by the following solutions

$$\mathbf{w}_1 = \begin{bmatrix} 0 \\ 3197/2400 \\ -2497/1500 \\ 0 \\ -19997/12000 \\ 31961/12000 \\ -997/3000 \\ 0 \\ -3997/12000 \\ 0 \end{bmatrix} \implies \beta\|\mathbf{w}_1\|_1 + \frac{1}{2n}\|\mathbf{A}_e\mathbf{w}_1 - \mathbf{y}\|_2^2 = \frac{1999}{2500000}$$

$$\mathbf{w}_2 = \begin{bmatrix} 0 \\ 191823/140000 \\ -990613/840000 \\ -471683/420000 \\ -128017/120000 \\ 367547/140000 \\ -127357/840000 \\ -87827/420000 \\ -31993/120000 \\ 0 \end{bmatrix} \implies \beta\|\mathbf{w}_2\|_1 + \frac{1}{2n}\|\mathbf{A}_e\mathbf{w}_2 - \mathbf{y}\|_2^2 = \frac{1999}{2500000},$$

where each solution satisfies the optimality conditions in (40). We also provide a visualization for the output functions of each solution in Figure 12, namely Solution1 and Solution2.

Remark 1. *In fact, there exist infinitely many solutions to the regularized training problem dis-*

33

cussed above, which can be analytically defined by the following weights and biases

$$\mathbf{u}^{*T} = [1\ 1\ 1\ 1\ 1\ -1\ -1\ -1\ -1\ -1], \ \mathbf{b}^{*T} = [2\ 1\ 0\ -c\ -1\ -1\ 0\ c\ 1\ 2]$$

where c can be arbitrarily chosen to satisfy $0 \le c \le 1$. As a numerical proof, we also provide two additional examples with $c = 0.2$ and $c = 0.8$, i.e., Solution3 and Solution4, in Figure 12. These solutions also achieve the same objective value and their last layer weights are as follows

$$\mathbf{w}_3 = \begin{bmatrix} 0 \\ 323691/248000 \\ -7349999/5208000 \\ -1039627/1041600 \\ -4660169/5208000 \\ 667193/248000 \\ -1810997/5208000 \\ -199753/1041600 \\ -795707/5208000 \\ 0 \end{bmatrix} \ and \ \mathbf{w}_4 = \begin{bmatrix} 0 \\ 167847/116000 \\ -1248409/1218000 \\ -1058987/974400 \\ -6500131/4872000 \\ 295631/116000 \\ -25387/1218000 \\ -99563/974400 \\ -2082883/4872000 \\ 0 \end{bmatrix}.$$

This numerical observation can also be explained via our extreme point characterization in (24), where the optimal bias can take one of infinitely many possible values in a certain interval when $\sum_{i \in \mathcal{S}} v_i = 0$.

$\square$

Proof of Corollary 3.4. Given $\mathbf{A} = \mathbf{c}\mathbf{a}^T$, all possible extreme points can be characterized as follows

$$\underset{b,\mathbf{u}:\|\mathbf{u}\|_2=1}{\operatorname{argmax}} |\mathbf{v}^T(\mathbf{A}\mathbf{u} + b\mathbf{1})_+| = \underset{b,\mathbf{u}:\|\mathbf{u}\|_2=1}{\operatorname{argmax}} |\mathbf{v}^T(\mathbf{c}\mathbf{a}^T\mathbf{u} + b\mathbf{1})_+|$$

$$= \underset{b,\mathbf{u}:\|\mathbf{u}\|_2=1}{\operatorname{argmax}} \left| \sum_{i=1}^{n} v_i (c_i\mathbf{a}^T\mathbf{u} + b)_+ \right|$$

which can be equivalently stated as

$$\underset{b,\mathbf{u}:\|\mathbf{u}\|_2=1}{\operatorname{argmax}} \sum_{i \in \mathcal{S}} v_i c_i \mathbf{a}^T\mathbf{u} + \sum_{i \in \mathcal{S}} v_i b \ \text{s.t.} \ \begin{cases} c_i\mathbf{a}^T\mathbf{u} + b \ge 0, \forall i \in \mathcal{S} \\ c_j\mathbf{a}^T\mathbf{u} + b \le 0, \forall j \in \mathcal{S}^c \end{cases},$$

which shows that $\mathbf{u}$ must be either positively or negatively aligned with $\mathbf{a}$, i.e., $\mathbf{u} = s\frac{\mathbf{a}}{\|\mathbf{a}\|_2}$, where $s = \pm 1$. Thus, b must be in the range of $[\max_{i \in \mathcal{S}}(-sc_i\|\mathbf{a}\|_2), \ \min_{j \in \mathcal{S}^c}(-sc_j\|\mathbf{a}\|_2)]$ Using these observations, extreme points can be formulated as follows

$$\mathbf{u}_v = \begin{cases} \frac{\mathbf{a}}{\|\mathbf{a}\|_2} & \text{if } \sum_{i \in \mathcal{S}} v_i c_i \ge 0 \\ \frac{-\mathbf{a}}{\|\mathbf{a}\|_2} & \text{otherwise} \end{cases} \ \text{and} \ b_v = \begin{cases} \min_{j \in \mathcal{S}^c}(-s_v c_j \|\mathbf{a}\|_2) & \text{if } \sum_{i \in \mathcal{S}} v_i \ge 0 \\ \max_{i \in \mathcal{S}}(-s_v c_i \|\mathbf{a}\|_2) & \text{otherwise} \end{cases},$$

where $s_v = \operatorname{sign}(\sum_{i \in \mathcal{S}} v_i c_i)$.

$\square$

Proof of Lemma 3.1. Since $\mathbf{y}$ has both positive and negative entries, we need at least two $\mathbf{u}$'s with positive and negative output weights to represent $\mathbf{y}$ using the output range of ReLU. Therefore

the optimal value of the ℓ_0 problem is at least 2. Note that $\mathbf{A}\mathbf{A}^\dagger = \mathbf{I}_n$ since $\mathbf{A}$ is full row rank. Then let us define the output weights

$$w_1 = \|\mathbf{A}^\dagger(\mathbf{y})_+\|_2$$
$$w_2 = -\|\mathbf{A}^\dagger(-\mathbf{y})_+\|_2 \,.$$

Then note that

$$
\begin{aligned}
w_1(\mathbf{A}\mathbf{u}_1)_+ + w_2(\mathbf{A}\mathbf{u}_2)_+ &= (\mathbf{A}\mathbf{A}^\dagger(\mathbf{y})_+)_+ - (\mathbf{A}\mathbf{A}^\dagger(-\mathbf{y})_+)_+ \\
&= ((\mathbf{y})_+)_+ - ((-\mathbf{y})_+)_+ \\
&= (\mathbf{y})_+ - (-\mathbf{y})_+ \\
&= \mathbf{y}
\end{aligned}
$$

where the second equality follows from $\mathbf{A}\mathbf{A}^\dagger = \mathbf{I}_n$. $\qquad\square$

Proof of Lemma 3.2. We first provide the optimality conditions for the convex program in the following proposition:

Proposition 1. *Let $\mathbf{U}$ be a weight matrix for [Main Paper, (4)]. Then, $\mathbf{U} \in \mathbb{R}^{d\times m}$ is an optimal solution for the regularized training problem if*

$$\exists \boldsymbol{\alpha} \in \mathbb{R}^n, \mathbf{w} \in \mathbb{R}^m \ s.t. \ (\mathbf{A}\mathbf{U})_+ \mathbf{w} = \mathbf{y}, \quad (\mathbf{A}\mathbf{U})_+^T \boldsymbol{\alpha} = sign(\mathbf{w}) \tag{41}$$

and

$$\max_{\mathbf{u}\,:\,\|\mathbf{u}\|_2\leq 1} |\boldsymbol{\alpha}^T(\mathbf{A}\mathbf{u})_+| \leq 1\,. \tag{42}$$

These conditions follow from linear semi-infinite optimality conditions given in Theorem 7.1 and 7.6 [35] for Farkas-Minkowski systems. Then the proof Lemma 3.2 directly follows from the solution of minimum cardinality problem given in Lemma 3.1.

Now we prove the second claim. For whitened data matrices, denoting the Singular Value Decomposition of the input data as $\mathbf{A} = \mathbf{U}$ where $\mathbf{U}^T\mathbf{U} = \mathbf{U}\mathbf{U}^T = \mathbf{I}_n$ since $\mathbf{A}$ is assumed full row rank. Consider the dual optimization problem

$$\max_{|\mathbf{v}^T(\mathbf{A}\mathbf{u})_+|\leq 1,\ \forall \mathbf{u}\in\mathcal{B}_2} \mathbf{v}^T\mathbf{y} \tag{43}$$

Changing the variable to $\mathbf{u}' = \mathbf{U}\mathbf{u}$ in the dual problem we next show that

$$\max_{|\mathbf{v}^T(\mathbf{u}')_+|\leq 1,\ \forall \mathbf{u}'\in\mathcal{B}_2} \mathbf{v}^T\mathbf{y} = \max_{\|(\mathbf{v})_+\|_2\leq 1,\ \|(-\mathbf{v})_+\|_2\leq 1} \mathbf{v}^T\mathbf{y}\,. \tag{44}$$

where the equality follows from the upper bound

$$\mathbf{v}^T(\mathbf{u}')_+ \leq (\mathbf{v})_+^T(\mathbf{u}')_+ \leq \|(\mathbf{v})_+\|_2\|(\mathbf{u}')_+\|_2 \leq \|(\mathbf{v})_+\|_2\,,$$

which is achieved when $\mathbf{u}' = \frac{(\mathbf{v})_+}{\|(\mathbf{v})_+\|_2}$. Similarly we have

$$-\mathbf{v}^T(\mathbf{u}')_+ \leq (-\mathbf{v})_+^T(\mathbf{u}')_+ \leq \|(-\mathbf{v})_+\|_2\|(\mathbf{u}')_+\|_2 \leq \|(-\mathbf{v})_+\|_2\,,$$

which is achieved when $\mathbf{u}' = \frac{(-\mathbf{v})_+}{\|(-\mathbf{v})_+\|_2}$, which verifies the right-hand-side of (44). Now note that

$$\mathbf{v}^T \mathbf{y} \le (\mathbf{v})_+^T (\mathbf{y})_+ + (-\mathbf{v})_+^T (-\mathbf{y})_+ \,.$$

Therefore the right-hand-side of (44) is upper-bounded by $\|(\mathbf{y})_+\|_2 + \|(-\mathbf{y})_+\|_2$. This upper-bound is achieved by the choice

$$\mathbf{v} = \frac{(\mathbf{y})_+}{\|(\mathbf{y})_+\|_2} - \frac{(-\mathbf{y})_+}{\|(-\mathbf{y})_+\|_2} \,,$$

since we have

$$\mathbf{v}^T \mathbf{y} = \frac{\mathbf{y}^T (\mathbf{y})_+}{\|(\mathbf{y})_+\|_2} - \frac{\mathbf{y}^T (-\mathbf{y})_+}{\|(-\mathbf{y})_+\|_2} = \frac{(\mathbf{y})_+^T (\mathbf{y})_+}{\|(\mathbf{y})_+\|_2} + \frac{(-\mathbf{y})_+^T (-\mathbf{y})_+}{\|(-\mathbf{y})_+\|_2}$$
$$= \|(\mathbf{y})_+\|_2 + \|(-\mathbf{y})_+\|_2 \,.$$

Therefore the preceding choice of $\mathbf{v}$ is optimal. Consequently, the corresponding optimal neuron weights satisfy

$$\mathbf{u}_1' = \frac{(\mathbf{y})_+}{\|(\mathbf{y})_+\|_2} \quad \text{and} \quad \mathbf{u}_2' = \frac{(-\mathbf{y})_+}{\|(-\mathbf{y})_+\|_2} \,.$$

Changing the variable back via $\mathbf{u} = \mathbf{U}^T \mathbf{u}' = \mathbf{A}^\dagger \mathbf{u}'$ we conclude that the optimal neurons are given by

$$\mathbf{u}_1 = \frac{\mathbf{A}^\dagger (\mathbf{y})_+}{\|(\mathbf{y})_+\|_2} \quad \text{and} \quad \mathbf{u}_2 = \frac{\mathbf{A}^\dagger (-\mathbf{y})_+}{\|(-\mathbf{y})_+\|_2} \,,$$

or equivalently

$$\mathbf{u}_1 = \frac{\mathbf{A}^\dagger (\mathbf{y})_+}{\|\mathbf{A}^\dagger (\mathbf{y})_+\|_2} \quad \text{and} \quad \mathbf{u}_2 = \frac{\mathbf{A}^\dagger (-\mathbf{y})_+}{\|\mathbf{A}^\dagger (-\mathbf{y})_+\|_2} \,,$$

since $\mathbf{A}^\dagger$ is orthonormal and yields the claimed expression. Finally, note that the corresponding output weights are $\|\mathbf{A}^\dagger (\mathbf{y})_+\|_2$ and $\|\mathbf{A}^\dagger (-\mathbf{y})_+\|_2$, respectively. $\qquad\square$

Proof of Proposition 3.2. Since the constraint in [Main Paper,(15)] is bounded below and the hidden layer weights are constrained to the unit Euclidean ball, the convergence of the cutting plane method directly follows from Theorem 11.2 of [35]. $\qquad\square$

Proof of Theorem 3.4. Given a vector $\mathbf{u}$ we partition $\mathbf{A}$ according to the subset $S = \{i | \mathbf{a}_i^T \mathbf{u} \ge 0\}$, where $\mathbf{A}_S \mathbf{u} \succcurlyeq 0$ and $-\mathbf{A}_{S^c} \mathbf{u} \succcurlyeq 0$ into

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_S \\ \mathbf{A}_{S^c} \end{bmatrix} \,.$$

Here $\mathbf{A}_S$ is the sub-matrix of $\mathbf{A}$ consisting of the rows indexed by S, and S^c is the complement of the set S. Consequently, we partition the vector $(\mathbf{Au})_+$ as follows

$$(\mathbf{Au})_+ = \begin{bmatrix} \mathbf{A}_S \mathbf{u} \\ \mathbf{0} \end{bmatrix} \,.$$

Then we use the block matrix pseudo-inversion formula [37]

$$\mathbf{A}^\dagger = \left[\begin{array}{cc} \left(\mathbf{A}_S \mathbf{P}_{S^c}^\perp\right)^\dagger & \left(\mathbf{A}_{S^c} \mathbf{P}_S^\perp\right)^\dagger \end{array} \right],$$

where $\mathbf{P}_S$ and $\mathbf{P}_{S^c}$ are projection matrices defined as follows

$$\mathbf{P}_S = \mathbf{I}_d - \mathbf{A}_S^T \left(\mathbf{A}_S \mathbf{A}_S^T\right)^{-1} \mathbf{A}_S$$
$$\mathbf{P}_{S^c} = \mathbf{I}_d - \mathbf{A}_{S^c}^T \left(\mathbf{A}_{S^c} \mathbf{A}_{S^c}^T\right)^{-1} \mathbf{A}_{S^c}.$$

Note that the matrices $\mathbf{A}_S \mathbf{A}_S^T \in \mathbb{R}^{|S| \times |S|}$, $\mathbf{A}_{S^c} \mathbf{A}_{S^c}^T \in \mathbb{R}^{|S^c| \times |S^c|}$ are full column rank with probability one since the matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$ is i.i.d. Gaussian where $n < d$. Hence the inverses $\left(\mathbf{A}_S \mathbf{A}_S^T\right)^{-1}$ and $\left(\mathbf{A}_{S^c} \mathbf{A}_{S^c}^T\right)^{-1}$ exist with probability one. Plugging in the above representation in the spike-free condition we get

$$\mathbf{A}^\dagger \left(\mathbf{A}\mathbf{u}\right)_+ = \left(\mathbf{A}_S \mathbf{P}_{S^c}^\perp\right)^\dagger \mathbf{A}_S \mathbf{u}.$$

Then we can express the probability of the matrix being spike-free as

$$\mathbb{P}\left[\max_{\mathbf{u} \in \mathcal{B}_2} \|\mathbf{A}^\dagger(\mathbf{A}\mathbf{u})_+\|_2 > 1\right] = \mathbb{P}\left[\exists \mathbf{u} \in \mathcal{B}_2 \mid \|\mathbf{A}^\dagger(\mathbf{A}\mathbf{u})_+\|_2 > 1\right]$$
$$\leq \mathbb{P}\left[\exists \mathbf{u} \in \mathcal{B}_2, S \subseteq [n] \mid \|\left(\mathbf{A}_S \mathbf{P}_{S^c}^\perp\right)^\dagger \mathbf{A}_S \mathbf{u}\|_2 > 1\right].$$

Finally, observe that $\mathbf{P}_{S^c} \in \mathbb{R}^{d \times d}$ is a uniformly random projection matrix of subspace of dimension $|S| \leq n$. Therefore as $d \to \infty$, we have $\mathbf{P}_{S^c}^\perp \to \mathbf{I}_d$, and consequently

$$\lim_{d \to \infty} \|\left(\mathbf{A}_S \mathbf{P}_{S^c}^\perp\right)^\dagger \mathbf{A}_S \mathbf{u}\|_2 = \|\mathbf{A}_S^\dagger \mathbf{A}_S \mathbf{u}\|_2,$$

with probability one, and we have

$$\lim_{d \to \infty} \mathbb{P}\left[\exists \mathbf{u} \in \mathcal{B}_2, S \subseteq [n] \mid \|\left(\mathbf{A}_S \mathbf{P}_{S^c}^\perp\right)^\dagger \mathbf{A}_S \mathbf{u}\|_2 > 1\right] = 0.$$

$\square$

Proof of Theorem 3.5. Since each sample $\mathbf{a}_j$ is a vertex of $\mathcal{C}_a$, we can find a separating hyperplane defined by the parameters $(\mathbf{u}_j, b_j)$ so that $\mathbf{a}_j^T \mathbf{u}_j + b_j > 0$ and $\mathbf{a}_i^T \mathbf{u}_j + b_j \leq 0, \forall i \neq j$. Then, choosing $\{(\mathbf{u}_j, b_j)\}_{j=1}^n$ yields that $(\mathbf{A}\mathbf{U} + \mathbf{1}\mathbf{b}^T)_+$ is a diagonal matrix. Using these hidden neurons, we write the constraint of [Main Paper,(4)] in a more compact form as

$$(\mathbf{A}\mathbf{U} + \mathbf{1}\mathbf{b}^T)_+ \mathbf{w} = \mathbf{y},$$

which is a least squares problem with a full rank square data matrix. Therefore, selecting $\mathbf{w} = ((\mathbf{A}\mathbf{U} + \mathbf{1}\mathbf{b}^T)_+)^\dagger \mathbf{y}$ along with $\mathbf{U}$ and $\mathbf{b}$ achieves a feasible solution for the original problem, i.e., 0 training error. $\square$

Proof of Theorem 3.6. Let us define the distance of the ith sample vector to the convex hull of the remaining sample vectors as d_i

$$d_i \triangleq \min_{\substack{\mathbf{z} \in \mathbb{R}^n: \\ \sum_{j \neq i} z_j = 1 \\ \mathbf{z} \succeq 0}} \|\mathbf{a}_i - \sum_{j \neq i} \mathbf{a}_j z_j\|_2 = \min_{\substack{\mathbf{z} \in \mathbb{R}^n: \\ \sum_{j \neq i} z_j = 1 \\ \mathbf{z} \succeq 0, z_i = -1}} \|\mathbf{A}^T \mathbf{z}\|_2$$

$$= \min_{\substack{\mathbf{z} \in \mathbb{R}^n: \\ \sum_{j \neq i} z_j = 1 \\ \mathbf{z} \succeq 0, z_i = -1}} \max_{\mathbf{v}: \|\mathbf{v}\|_2 \leq 1} \mathbf{v}^T \mathbf{A}^T \mathbf{z}$$

Using Gordon's escape from a mesh theorem [38, 39], we obtain the following lower-bound on the expectation of d_i

$$\mathbb{E}\, d_i \geq \mathbb{E} \min_{\substack{\mathbf{z} \in \mathbb{R}^n: \\ \sum_{j \neq i} z_j = 1 \\ z_j \geq 0, z_i = -1}} \max_{\mathbf{v}: \|\mathbf{v}\|_2 \leq 1} \mathbf{h}^T \mathbf{v} \|\mathbf{z}\|_2 + \mathbf{z}^T \mathbf{g}$$

$$= \mathbb{E} \min_{\substack{\mathbf{z} \in \mathbb{R}^n: \\ \sum_{j \neq i} z_j = 1 \\ z_j \geq 0, z_i = -1}} \|\mathbf{h}\|_2 \|\mathbf{z}\|_2 + \mathbf{z}^T \mathbf{g}$$

$$\geq \sqrt{d} \|\mathbf{z}\|_2 - \mathbb{E} \max_{j \in [n], j \neq i} g_j + g_i$$

$$\geq \frac{\sqrt{d}}{\sqrt{n}} - \sqrt{2 \log(n-1)}, \tag{45}$$

where $\mathbf{h} \in \mathbb{R}^d$ and $\mathbf{g} \in \mathbb{R}^n$ are random vectors with i.i.d. standard Gaussian components, and the second inequality follows from a well-known result on finite Gaussian suprema [39]. Therefore, the expected distance of the ith sample to the convex hull is guaranteed to be positive whenever $d > 2n \log(n-1)$. Note that the lower bound (45) is vacuous for $d < 2n \log(n-1)$ since the random variable d_i can only take non-negative values.

The distance d_i is a Lipschitz function of the random Gaussian matrix $\mathbf{A}$. This can be seen via the following argument

$$\min_{\substack{\mathbf{z} \in \mathbb{R}^n: \\ \sum_{j \neq i} z_j = 1 \\ z_j \geq 0, z_i = -1}} \|\mathbf{A}^T \mathbf{z}\|_2 - \min_{\substack{\mathbf{z} \in \mathbb{R}^n: \\ \sum_{j \neq i} z_j = 1 \\ z_j \geq 0, z_i = -1}} \|\tilde{\mathbf{A}}^T \mathbf{z}\|_2 \leq \min_{\substack{\mathbf{z} \in \mathbb{R}^n: \\ \sum_{j \neq i} z_j = 1 \\ z_j \geq 0, z_i = -1}} \|(\mathbf{A} - \tilde{\mathbf{A}}^T) \mathbf{z}\|_2$$

$$\leq \|(\mathbf{A} - \tilde{\mathbf{A}})\|_2 \max_{\substack{\mathbf{z} \in \mathbb{R}^n: \\ \sum_{j \neq i} z_j = 1 \\ z_j \geq 0, z_i = -1}} \|\mathbf{z}\|_2$$

$$\leq \|(\mathbf{A} - \tilde{\mathbf{A}})\|_F \max_{\substack{\mathbf{z} \in \mathbb{R}^n: \\ \sum_{j \neq i} z_j = 1 \\ z_j \geq 0, z_i = -1}} \|\mathbf{z}\|_1$$

$$\leq 2 \|(\mathbf{A} - \tilde{\mathbf{A}})\|_F$$

Applying the Lipschitz concentration for Gaussian measure [39] yields that

$$\mathbb{P}\big[d_i > \sqrt{d} - \sqrt{2n \log(n-1)} - t\big] \geq 1 - 2e^{-t^2/2}.$$

Therefore, we have $d_i > 0$ for $d > 2n \log(n-1)$ with probability exceeding $1 - 2e^{-t^2/2}$. Taking a union bound over every index $i \in \{0, ..., n\}$, we can upper-bound the failure probability by $2ne^{-t^2/2}$. Choosing $t^2 = 4 \log(n-1)$ will yield a failure probability $O(1/n)$ and conclude the proof. $\square$

Proof of Theorem 4.1. The proof follows from a similar argument as in the proof of Theorem 3.1. We reparameterize [Main Paper,(16)] as follows

$$\min_{\mathbf{r},\theta\in\Theta} \frac{1}{2}\|\mathbf{r}\|_2^2 + \beta\|\mathbf{w}\|_1 \text{ s.t. } \mathbf{r} = (\mathbf{AU})_+\mathbf{w} - \mathbf{y}, \ \|\mathbf{u}_j\|_2 \le 1, \forall j.$$

Then taking the dual of the above problem yields the claimed form. $\qquad\square$

Proof of Theorem 5.1. The proof follows from a similar argument as in the proof of Theorem 3.1 and 4.1. We first put [Main Paper,(18)] into the following form

$$\min_{\xi,\theta\in\Theta} \sum_{i=1}^{n} \xi_i + \beta\|\mathbf{w}\|_1 \text{ s.t. } \xi_i \ge 0, \ \xi_i \ge 1 - y_i(\mathbf{a}_i^T\mathbf{U})_+\mathbf{w}, \forall i, \ \|\mathbf{u}_j\|_2 \le 1, \forall j.$$

Then taking the dual of the above problem yields the claimed form. $\qquad\square$

Proof of Theorem 6.1. The proof follows from classical Fenchel duality [40], and a similar argument as in the proof of Theorem 3.1 and 4.1. We first describe [Main Paper,(19)] in an equivalent form as follows

$$\min_{\mathbf{z},\theta\in\Theta} \ell(\mathbf{z},\mathbf{y}) + \beta\|\mathbf{w}\|_1 \text{ s.t. } \mathbf{z} = (\mathbf{AU})_+\mathbf{w}, \ \|\mathbf{u}_j\|_2 \le 1, \forall j.$$

Then the dual function is

$$g(\mathbf{v}) = \min_{\mathbf{z},\theta\in\Theta} \ell(\mathbf{z},\mathbf{y}) - \mathbf{v}^T\mathbf{z} + \mathbf{v}^T(\mathbf{AU})_+\mathbf{w} + \beta\|\mathbf{w}\|_1 \text{ s.t. } \|\mathbf{u}_j\|_2 \le 1, \forall j.$$

Therefore, using the classical Fenchel duality [40] yields the proposed dual form. $\qquad\square$

A.6   Polar convex duality

In this section we derive the polar duality and present a connection to minimum ℓ_1 solutions to linear systems. Recognizing the constraint $\mathbf{v} \in \mathcal{Q}_\mathbf{A}$ can be stated as

$$\mathbf{v} \in \mathcal{Q}_\mathbf{A}^\circ, \ \mathbf{v} \in -\mathcal{Q}_\mathbf{A}^\circ,$$

which is equivalent to

$$\mathbf{v} \in \mathcal{Q}_\mathbf{A}^\circ \cap -\mathcal{Q}_\mathbf{A}^\circ.$$

Note that the support function of a set can be expressed as the gauge function of its polar set (see e.g. [14]). The polar set of $\mathcal{Q}_\mathbf{A}^\circ \cap -\mathcal{Q}_\mathbf{A}^\circ$ is given by

$$\left(\mathcal{Q}_\mathbf{A}^\circ \cap -\mathcal{Q}_\mathbf{A}^\circ\right)^\circ = \text{conv}\{\mathcal{Q}_\mathbf{A} \cup -\mathcal{Q}_\mathbf{A}\}.$$

Using this fact, we express the dual problem [Main Paper, (12)] as

$$D^* = \inf_{t\in\mathbb{R}} t \tag{46}$$
$$\text{s.t. } \mathbf{y} \in t \text{ conv}\left\{\mathcal{Q}_\mathbf{A} \cup -\mathcal{Q}_\mathbf{A}\right\},$$

where conv represents the convex hull of a set.

Let us restate dual of the two-layer ReLU NN training problem given by

$$\max_{\mathbf{v}} \mathbf{v}^T\mathbf{y} \text{ s.t. } \mathbf{v} \in \mathcal{Q}_\mathbf{A}^\circ, \ -\mathbf{v} \in \mathcal{Q}_\mathbf{A}^\circ \tag{47}$$

where $\mathcal{Q}_\mathbf{A}^\circ$ is the polar dual of $\mathcal{Q}_\mathbf{A}$ defined as $\mathcal{Q}_\mathbf{A}^\circ = \{\mathbf{v}|\mathbf{v}^T\mathbf{u} \le 1 \forall \mathbf{u} \in \mathcal{Q}_\mathbf{A}\}$.

Remark 2. *The dual problem given in (47) is analogous to the convex duality in minimum ℓ_1 norm solutions to linear systems. In particular, for the latter it holds that*

$$\min_{\mathbf{w}\,:\,\mathbf{Aw}=\mathbf{y}} \|\mathbf{w}\|_1 = \max_{\mathbf{v}\in\mathrm{conv}\{\hat{\mathbf{a}}_1,...,\hat{\mathbf{a}}_d\}^\circ,\ -\mathbf{v}\in\mathrm{conv}\{\hat{\mathbf{a}}_1,...,\hat{\mathbf{a}}_d\}^\circ} \mathbf{v}^T\mathbf{y}\,,$$

where $\hat{\mathbf{a}}_1,...,\hat{\mathbf{a}}_d$ are the columns of $\mathbf{A}$. The above optimization problem can also be put in the gauge optimization form as follows.

$$\min_{\mathbf{w}\,:\,\mathbf{Aw}=\mathbf{y}} \|\mathbf{w}\|_1 = \inf_{t\in\mathbb{R}} t \ \text{s.t.} \ \mathbf{y} \in t\,\mathrm{conv}\{\pm\hat{\mathbf{a}}_1,...,\pm\hat{\mathbf{a}}_d\},$$

which parallels the gauge optimization form in (46).

A.7 Extension to vector output neural networks

In this section, we first derive the dual form for vector output NN and then describe the implementation of the cutting plane algorithm.

Here, we have $\mathbf{Y} \in \mathbb{R}^{n\times o}$ and $f(\mathbf{A}) = \left(\mathbf{AU}\right)_+\mathbf{W}$, where $\mathbf{W} \in \mathbb{R}^{m\times o}$. Then, we have the following problem

$$\min_{\theta\in\Theta} \|\mathbf{W}\|_F^2 + \|\mathbf{U}\|_F^2 \ \text{s.t.} \ \left(\mathbf{AU}\right)_+\mathbf{W} = \mathbf{Y}.$$

Lemma 1. *The following two optimization problems are equivalent:*

$$\min_{\theta\in\Theta} \|\mathbf{W}\|_F^2 + \|\mathbf{U}\|_F^2 \ \textit{s.t.} \ \left(\mathbf{AU}\right)_+\mathbf{W} = \mathbf{Y} \quad = \quad \min_{\theta\in\Theta} \sum_{j=1}^{m} \|\mathbf{w}_j\|_2 \ \textit{s.t.} \ \left(\mathbf{AU}\right)_+\mathbf{W} = \mathbf{Y}, \ \|\mathbf{u}_j\|_2^2 \le 1, \ \forall j$$

Proof. For any $\theta \in \Theta$, we can rescale the parameters as $\bar{\mathbf{u}}_j = \alpha_j\mathbf{u}_j$ and $\bar{\mathbf{w}}_j = \mathbf{w}_j/\alpha_j$, for any $\alpha_j > 0$, where $\mathbf{u}_j$ and $\mathbf{w}_j$ are the jth column and row of $\mathbf{U}$ and $\mathbf{W}$, respectively. Then, [Main Paper, (1)] becomes

$$f_{\bar\theta}(\mathbf{A}) = \sum_{j=1}^{m}(\mathbf{A}\bar{\mathbf{u}}_j)_+\bar{\mathbf{w}}_j^T = \sum_{j=1}^{m}(\alpha_j\mathbf{A}\mathbf{u}_j)_+\frac{\mathbf{w}_j^T}{\alpha_j} = \sum_{j=1}^{m}(\mathbf{A}\mathbf{u}_j)_+\mathbf{w}_j^T,$$

which proves $f_\theta(\mathbf{A}) = f_{\bar\theta}(\mathbf{A})$. In addition to this, we have the following basic inequality

$$\sum_{j=1}^{m}(\|\mathbf{w}_j\|_2^2 + \|\mathbf{u}_j\|_2^2) \ge 2\sum_{j=1}^{m}(\|\mathbf{w}_j\|_2 \, \|\mathbf{u}_j\|_2),$$

where the equality is achieved with the scaling choice $\alpha_j = \left(\frac{\|\mathbf{w}_j\|_2}{\|\mathbf{u}_j\|_2}\right)^{\frac{1}{2}}$. Since the scaling operation does not change the right-hand side of the inequality, we can set $\|\mathbf{u}_j\|_2 = 1, \forall j$. Therefore, the right-hand side becomes $\sum_{j=1}^{m}\|\mathbf{w}_j\|_2$. $\qquad\square$

Using Lemma 1, we get the following equivalent form

$$\min_{\theta\in\Theta} \sum_{j=1}^{m} \|\mathbf{w}_j\|_2 \ \text{s.t.} \ \left(\mathbf{AU}\right)_+\mathbf{W} = \mathbf{Y}, \ \|\mathbf{u}_j\|_2^2 \le 1, \ \forall j, \tag{48}$$

which has the following dual form

$$\max_{\mathbf{V}} \mathrm{trace}(\mathbf{V}^T\mathbf{Y}) \ \text{s.t.} \ \|\mathbf{V}^T(\mathbf{Au}_j)_+\|_2 \le 1, \ \|\mathbf{u}_j\|_2 \le 1, \ \forall j. \tag{49}$$

Then, we again relax the problem using the spike-free relaxation and then we solve the following problem to achieve the extreme points

$$\hat{\mathbf{u}} = \operatorname*{argmax}_{\mathbf{u}} \|\mathbf{V}^T \mathbf{A} \mathbf{u}\|_2 \text{ s.t. } \mathbf{A}\mathbf{u} \succcurlyeq \mathbf{0}, \|\mathbf{u}\|_2 \leq 1 \tag{50}$$

Thus, the hidden layers can be determined by solving the above optimization problem.

A.7.1 Solutions to one dimensional problems

Here, we consider a vector output problem with a one dimensional data matrix, i.e., $\mathbf{A} = \mathbf{a}$, where $\mathbf{a} \in \mathbb{R}^n$. Then, the extreme points of (49) can be obtained via the following maximization problem

$$\operatorname*{argmax}_{u,b} \|\mathbf{V}^T(\mathbf{a}u + b\mathbf{1})_+\|_2^2 \text{ s.t. } |u| = 1. \tag{51}$$

Using the same steps in Lemma 2.6, we can write (51) as follows

$$\operatorname*{argmax}_{u,b} \left\| \sum_{i \in \mathcal{S}} \mathbf{v}_i(a_i u + b) \right\|_2^2 \text{ s.t. } a_i u + b \geq 0, \forall i \in \mathcal{S}, a_j u + b \leq 0, \forall j \in \mathcal{S}^c, |u| = 1. \tag{52}$$

Notice that u can be either $+1$ or -1. Thus, we can solve the problem separately and then pick the one with higher objective value. First assume that $u = +1$, then (52) becomes

$$\operatorname*{argmax}_{b} \left\| \sum_{i \in \mathcal{S}} \mathbf{v}_i(a_i + b) \right\|_2^2 \text{ s.t. } \max_{i \in \mathcal{S}} -a_i \leq b \leq \min_{j \in \mathcal{S}^c} -a_j. \tag{53}$$

Since

$$\left\| \sum_{i \in \mathcal{S}} \mathbf{v}_i(a_i + b) \right\|_2^2 = \left\| \sum_{i \in \mathcal{S}} \mathbf{v}_i a_i \right\|_2^2 + 2b \left(\sum_{i \in \mathcal{S}} \mathbf{v}_i a_i \right)^T \sum_{i \in \mathcal{S}} \mathbf{v}_i + b^2 \left\| \sum_{i \in \mathcal{S}} \mathbf{v}_i \right\|_2^2,$$

(53) is a convex function of b. Therefore, the optimal solution to (53) is achieved when either $b = \min_{j \in \mathcal{S}^c} -a_j$ or $b = \max_{i \in \mathcal{S}} -a_i$ holds. Similar arguments also hold for $u = -1$.

Corollary 1. *Let $\{a_i\}_{i=1}^n$ be a one dimensional training set i.e., $a_i \in \mathbb{R}, \forall i \in [n]$. Then, the solutions to (48) that achieve the optimal value are extreme points, and therefore satisfy $\{(u_i, b_i)\}_{i=1}^m$, where $u_i = \pm 1, b_i = -sign(u_i)a_i$.*

Proof. The proof directly follows from the proof of Corollary 3.2. $\square$

A.7.2 Solutions to rank one problems

Here, we consider a vector output problem with a rank-one data matrix, i.e., $\mathbf{A} = \mathbf{c}\mathbf{a}^T$, all possible extreme points can be characterized as follows

$$\operatorname*{argmax}_{b,\mathbf{u}:\|\mathbf{u}\|_2=1} \left\| \mathbf{V}^T(\mathbf{A}\mathbf{u} + b\mathbf{1})_+ \right\|_2^2 = \operatorname*{argmax}_{b,\mathbf{u}:\|\mathbf{u}\|_2=1} \left\| \mathbf{V}^T(\mathbf{c}\mathbf{a}^T\mathbf{u} + b\mathbf{1})_+ \right\|_2^2$$

$$= \operatorname*{argmax}_{b,\mathbf{u}:\|\mathbf{u}\|_2=1} \left\| \sum_{i=1}^n \mathbf{v}_i(c_i \mathbf{a}^T \mathbf{u} + b)_+ \right\|_2^2$$

which can be equivalently stated as

$$\underset{b,\mathbf{u}:\|\mathbf{u}\|_2=1}{\operatorname{argmax}} \Big\| \sum_{i\in\mathcal{S}} \mathbf{v}_i c_i \Big\|_2^2 (\mathbf{a}^T\mathbf{u})^2 + 2b(\mathbf{a}^T\mathbf{u})\Big(\sum_{i\in\mathcal{S}} \mathbf{v}_i c_i \Big)^T \sum_{i\in\mathcal{S}} \mathbf{v}_i + b^2 \Big\| \sum_{i\in\mathcal{S}} \mathbf{v}_i \Big\|_2^2$$

$$\text{s.t.} \begin{cases} c_i\mathbf{a}^T\mathbf{u} + b \geq 0, \forall i \in \mathcal{S} \\ c_j\mathbf{a}^T\mathbf{u} + b \leq 0, \forall j \in \mathcal{S}^c \end{cases},$$

which shows that $\mathbf{u}$ must be either positively or negatively aligned with $\mathbf{a}$, i.e., $\mathbf{u} = s\frac{\mathbf{a}}{\|\mathbf{a}\|_2}$, where $s = \pm 1$. Thus, b must be in the range of $[\max_{i\in\mathcal{S}}(-sc_i\|\mathbf{a}\|_2), \ \min_{j\in\mathcal{S}^c}(-sc_j\|\mathbf{a}\|_2)]$ Using these observations, extreme points can be formulated as follows

$$\mathbf{u}_v = \begin{cases} \frac{\mathbf{a}}{\|\mathbf{a}\|_2} & \text{if } \sum_{i\in\mathcal{S}} v_i c_i \geq 0 \\ \frac{-\mathbf{a}}{\|\mathbf{a}\|_2} & \text{otherwise} \end{cases} \text{ and } b_v = \begin{cases} \min_{j\in\mathcal{S}^c}(-s_v c_j\|\mathbf{a}\|_2) & \text{if } \sum_{i\in\mathcal{S}} v_i \geq 0 \\ \max_{i\in\mathcal{S}}(-s_v c_i\|\mathbf{a}\|_2) & \text{otherwise} \end{cases},$$

where $s_v = \operatorname{sign}(\sum_{i\in\mathcal{S}} v_i c_i)$.

Corollary 2. *Let $\mathbf{A}$ be a data matrix such that $\mathbf{A} = \mathbf{c}\mathbf{a}^T$, where $\mathbf{c} \in \mathbb{R}^n$ and $\mathbf{a} \in \mathbb{R}^d$. Then, the solutions to (48) that achieve the optimal value are extreme points, and therefore satisfy $\{(\mathbf{u}_i, b_i)\}_{i=1}^m$, where $\mathbf{u}_i = s_i\frac{\mathbf{a}}{\|\mathbf{a}\|_2}, b_i = -s_i c_i\|\mathbf{a}\|_2$ with $s_i = \pm 1, \forall i \in [m]$.*

Proof. The proof directly follows from the proof of Corollary 3.4. $\qquad\square$

Theorem 1. *For one dimensional and/or rank-one datasets, solving the following ℓ_2-norm convex optimization problem globally optimizes (48)*

$$\min_{\mathbf{W}} \sum_{j=1}^m \|\mathbf{w}_j\|_2 \ s.t. \ \big(\mathbf{A}\mathbf{U}_e\big)_+\mathbf{W} = \mathbf{Y},$$

where $\mathbf{U}_e \in \mathbb{R}^{d\times m_e}$ is a weight matrix consisting of all possible extreme points.

Proof. Proof directly follows from Corollary 1 and 2. $\qquad\square$

A.7.3 ℓ_1 regularized version of vector output case

Notice that since (50) is a non-convex problem, finding extreme points in general is computationally expensive especially when the data dimensionality is high. Therefore, in this section, we provide an ℓ_1 regularized version of the problem in (48) so that extreme points can be efficiently achieved using convex optimization tools. Consider the following optimization problem

$$\min_{\theta\in\Theta} \|\mathbf{U}\|_F^2 + \sum_{j=1}^m \|\mathbf{w}_j\|_1^2 \ \text{s.t.} \ \big(\mathbf{A}\mathbf{U}\big)_+\mathbf{W} = \mathbf{Y}.$$

Then using the scaling trick in Lemma 2.1, we obtain the following

$$\min_{\theta\in\Theta} \sum_{j=1}^m \|\mathbf{w}_j\|_1 \ \text{s.t.} \ \big(\mathbf{A}\mathbf{U}\big)_+\mathbf{W} = \mathbf{Y}, \ \|\mathbf{u}_j\|_2^2 \leq 1, \ \forall j,$$

which has the following dual form

$$\max_{\mathbf{V}} \operatorname{trace}(\mathbf{V}^T\mathbf{Y}) \ \text{s.t.} \ \|\mathbf{V}^T\big(\mathbf{A}\mathbf{u}\big)_+\|_\infty \leq 1, \forall\mathbf{u} \in \mathcal{B}_2$$

and an optimal $\mathbf{U}$ satisfies

$$\|(\mathbf{A}\mathbf{U}^*)_+^T\mathbf{V}^*\|_\infty = 1\,,$$

where $\mathbf{V}^*$ is the optimal dual variable. Note that we can also consider block ℓ_1-ℓ_2 norms and their duals in formulating the vector output objective. We use this particular form as it admits a simpler solution with the cutting-plane method.

We again relax the problem using the spike-free relaxation and then we solve the following problem for each $k \in [o]$

$$\hat{\mathbf{u}}_{k,1} = \underset{\mathbf{u}}{\operatorname{argmax}}\, \mathbf{v}_k^T \mathbf{A}\mathbf{u} \text{ s.t. } \mathbf{A}\mathbf{u} \succcurlyeq \mathbf{0}, \|\mathbf{u}\|_2 \le 1$$

$$\hat{\mathbf{u}}_{k,2} = \underset{\mathbf{u}}{\operatorname{argmin}}\, \mathbf{v}_k^T \mathbf{A}\mathbf{u} \text{ s.t. } \mathbf{A}\mathbf{u} \succcurlyeq \mathbf{0}, \|\mathbf{u}\|_2 \le 1,$$

where $\mathbf{v}_k$ is the kth column of $\mathbf{V}$. After solving these optimization problems, we select the two neurons that achieve the maximum and minimum objective value among o neurons for each problem. Thus, we can find the weights for the hidden layers using convex optimization.

Consider the minimal cardinality problem

$$\min_{\theta \in \Theta} \|\mathbf{W}\|_0 \text{ s.t. } f_\theta(\mathbf{A}) = \mathbf{Y}, \|\mathbf{u}_j\|_2 = 1, \forall j.$$

The following result provides a characterization of the optimal solutions to the above problem

Lemma 2. *Suppose that $n \le d$, $\mathbf{A}$ is full row rank, and $\mathbf{Y} \in \mathbb{R}_+^{n \times o}$, e.g., one hot encoded outputs for multiclass classification and we have at least one sample in each class. Then an optimal solution to [Main Paper, (13)] is given by*

$$\mathbf{u}_k = \frac{\mathbf{A}^\dagger(\mathbf{y}_k)_+}{\|\mathbf{A}^\dagger(\mathbf{y}_k)_+\|_2} \text{ and } \mathbf{w}_k = \|\mathbf{A}^\dagger(\mathbf{y}_k)_+\|_2 \mathbf{e}_k$$

for each $k \in [o]$, where $\mathbf{w}_k$ and $\mathbf{y}_k$ are the kth row and column of $\mathbf{W}$ and $\mathbf{Y}$, respectively.

Proof. The proof is a straightforward generalization of the scalar output case in Lemma 3.1. $\qquad\square$

Lemma 3. *We have ℓ_1-ℓ_0 equivalence if the following condition holds*

$$\min_{\mathbf{v}:\mathbf{v}^T(\mathbf{A}\mathbf{u}_k)_+=1,\forall k} \; \max_{\mathbf{u}:\mathbf{u}\in\mathcal{B}_2} \mathbf{v}^T(\mathbf{A}\mathbf{u})_+ \le 1\,.$$

Proof. The proof is a straightforward generalization of the scalar output case in Lemma 3.2. $\qquad\square$