Journal of Machine Learning Research 22 (2021) 1-63 Submitted 12/20; Revised 7/21; Published 8/21

Convex Geometry and Duality of Over-parameterized Neural Networks

Tolga Ergen ERGEN@STANFORD.EDU
Department of Electrical Engineering
Stanford University
Stanford, CA 94305, USA

Mert Pilanci PILANCI@STANFORD.EDU
Department of Electrical Engineering
Stanford University
Stanford, CA 94305, USA

Editor: Julien Mairal

Abstract

We develop a convex analytic approach to analyze finite width two-layer ReLU networks. We first prove that an optimal solution to the regularized training problem can be characterized as extreme points of a convex set, where simple solutions are encouraged via its convex geometrical properties. We then leverage this characterization to show that an optimal set of parameters yield linear spline interpolation for regression problems involving one dimensional or rank-one data. We also characterize the classification decision regions in terms of a kernel matrix and minimum ℓ_1-norm solutions. This is in contrast to Neural Tangent Kernel which is unable to explain predictions of finite width networks. Our convex geometric characterization also provides intuitive explanations of hidden neurons as auto-encoders. In higher dimensions, we show that the training problem can be cast as a finite dimensional convex problem with infinitely many constraints. Then, we apply certain convex relaxations and introduce a cutting-plane algorithm to globally optimize the network. We further analyze the exactness of the relaxations to provide conditions for the convergence to a global optimum. Our analysis also shows that optimal network parameters can be also characterized as interpretable closed-form formulas in some practically relevant special cases.

Keywords: Neural Networks, ReLU Activation, Overparameterized Models, Convex Geometry, Duality

1. Introduction

Over-parameterized Deep Neural Networks (DNNs) have attracted significant attention due to their powerful representation and generalization capabilities. Recent studies empirically observed that NNs with ReLU activation achieve simple solutions as a result of training (see e.g., Maennel et al. (2018); Savarese et al. (2019)), although a full theoretical understanding is yet to be developed. Particularly, (Savarese et al., 2019) showed that among two-layer ReLU networks that perfectly fit one dimensional training data, i.e., $d = 1$, the minimum Euclidean norm ReLU network is a linear spline interpolator. Therefore, training over-parameterized networks with standard weight decay induces a bias towards simple solutions,

which may result in good generalization performance for $d = 1$. This work later extended to multi-variate functions ($d > 1$) in Ongie et al. (2020), however, the later work lacks an explicit characterization for the optimal solutions. Therefore, in the general case $d > 1$, characterizing the structure of optimal solutions and understanding the fundamental mechanism behind this implicit bias remain an open problem.

In this paper, we develop a convex analytic framework to reveal a fundamental convex geometric mechanism behind the bias towards simple solutions. More specifically, we show that over parameterized networks achieve simple solutions as the *extreme points* of a certain convex set, where simplicity is enforced by an implicit regularizer analogous to ℓ_1-norm regularization that promotes sparsity through extreme points of the unit ℓ_1-ball, i.e., the cross-polytope. However, unlike the conventional ℓ_1-norm regularization, extreme points are data-adaptive and can be interpreted as *convex autoenconders*. In the paper, we provide a complete characterization for extreme points via exact analytical expressions. As a corollary, for one dimensional and rank-one regression and classifications tasks, we prove that extreme points are in a specific form that yields linear spline interpolations, which explains the recent empirical observations in the $d = 1$ case. We also extend this analysis to higher dimensions ($d > 1$) to obtain exact characterizations or even closed-form solutions for the network parameters in some practically relevant cases.

1.1 Related work

Maennel et al. (2018); Blanc et al. (2019); Zhang et al. (2016) previously studied the dynamics of ReLU networks with finite neurons. Zhang et al. (2016) specifically showed that NNs are implicitly regularized so that training with Stochastic Gradient Descent (SGD) converges to small norm solutions. Later, Blanc et al. (2019) further elaborated the previous studies and proved that in a one dimensional case, SGD finds a solution that is linear over any set of three or more co-linear data points. Additionally, Maennel et al. (2018) proved that initialization magnitude of network parameters has a strong connection with implicit regularization. The authors further showed that in the regime where implicit regularization is effective, i.e., when initialization magnitude is small, network parameters align along certain directions characterized by the input data points. This observation shows that in fact there exist finitely many simple (or regularized) functions for a given training dataset. Chizat and Bach (2018) then proved that ReLU networks converge to a point that generalizes when initialization magnitude is small, i.e., called active training. Otherwise, parameters do not tend to vary and stay very close their initialization so that network does not generalize as well as in the small initialization case, which is also known as lazy training.

Another line of research in Bengio et al. (2006); Wei et al. (2018); Bach (2017); Chizat and Bach (2018) studied infinitely wide two-layer ReLU networks. In particular, Bengio et al. (2006) introduced a convex algorithm to train infinite width two-layer NNs. Although infinite dimensional training problems are not practical in higher dimensions, the analysis may shed light into the generalization properties. Wei et al. (2018) proved that over-parameterization improves generalization bounds by analyzing weakly regularized NNs. In addition to this, recently, the connection between infinite width NNs and kernel methods has attracted significant attention (Jacot et al., 2018; Arora et al., 2019). Such kernel based methods, nowadays known as Neural Tangent Kernel (NTK), work in a regime where

parameters barely change after initialization, coined 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.

Convexity of infinitely wide two-layer networks and kernel approximation is attractive due to the analytical tractability of convex optimization and tools from convex geometry, although these characterizations fall short of explaining the practical success of finite width networks. In Ergen and Pilanci (2019a); Bartan and Pilanci (2019) convex relaxations of one layer ReLU networks were studied, which have approximation guarantees under certain data assumptions. These architectures have a limited representation power due to the lack of a second layer.

1.2 Our contributions

Our contributions can be summarized as follows.

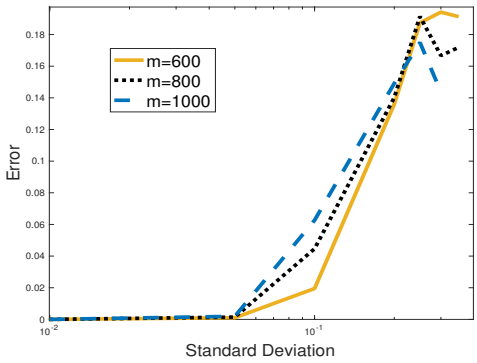
- We develop a convex analytic framework for two-layer ReLU NNs with weight decay, i.e., ℓ_2^2 regularization to provide a deeper insight into over-parameterization and implicit regularization. We prove that over-parameterized ReLU NNs behave like convex regularizers, which encourage simple solutions as the extreme points of a convex set which is termed *rectified ellipsoids*. The polar dual of a rectified ellipsoid is a convex body that determines the optimal hidden layer weights in a similar spirit to the extreme points of an ℓ_1-ball and its polar dual ℓ_∞-ball. We further show that the rectified ellipsoid is a data-dependent regularizer whose extreme points act as autoencoders (Lemma 7 and Lemma 8).

- As a corollary of our analysis, we show that optimal NNs that perfectly fit the training data outputs a linear spline interpolation for one dimensional or rank-one data. We also derive a general characterization for the hidden layer weights in higher dimensions in terms of a representer theorem to develop convex geometric insights (Corollary 1).

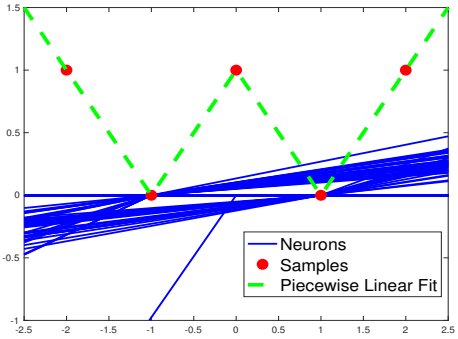
- Using our convex analytic framework, we characterize the set of optimal solutions in some specific cases so that the training problem can be reformulated as a convex optimization problem. We also show that there can be multiple globally optimal NNs with minimal ℓ_2^2 regularization, but leading to different predictions in these cases (Proposition 1 and Figure 7).

- For whitened data matrices, we provide exact closed form expressions of the optimal first and second layer weights by leveraging the convex duality (Theorems 9, 12 and 15). These expressions exhibit an interesting thresholding effect in a similar spirit to soft-thresholding of ℓ_1 penalty.

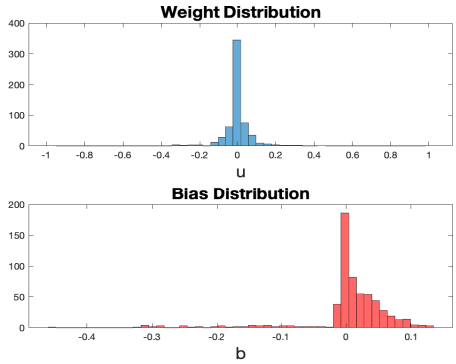
- Based on our convex analytic description, we propose training algorithms relying on convex relaxations of the rectified ellipsoid set. We further prove that the relaxations are tight in certain regimes when a convex geometric condition holds, including whitened and i.i.d random training data, and the algorithm globally optimizes the network.



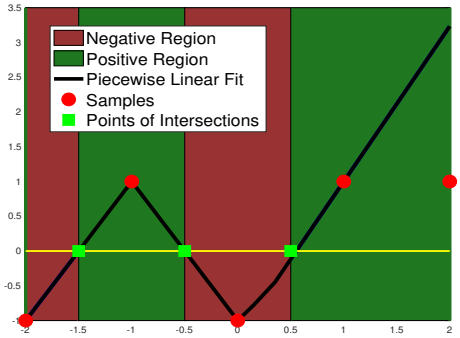
(a) Deviation of the ReLU network output from piecewise linear spline vs standard deviation of initialization plotted for different number of hidden neurons m.



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



(c) Weight and bias distributions for the network in Figure 1b.



(d) Binary classification using hinge loss. Network output is a linear spline interpolation, and decision regions are determined by zero crossings (see Lemma 6).

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

- We establish a connection between ℓ_0-ℓ_1 equivalence in compressed sensing and the training problem for ReLU networks (Lemma 10). Using this connection, we then obtain closed-form solutions for the optimal ReLU network parameters in certain practically relevant cases.

- We leverage our convex analytic characterization to design convex optimization based training methods that perform well in standard datasets and validate our theoretical results. In contrast to standard non-convex training methods, our methods provide transparent and interpretable means to train neural models.

1.3 Overview of our results

In order to understand the effects of initialization magnitude on implicit regularization, we train a two-layer ReLU network on one dimensional training data depicted in Figure 1b with different initialization magnitudes. In Figure 1a, we observe that the resulting optimal network output is linear spline interpolation when the initialization magnitude (i.e., the standard deviation of random initializations) is small, which matches with the empirical observations in recent work Maennel et al. (2018); Chizat and Bach (2018). We also provide the function fit by each neuron in Figure 1b in the case of small initialization magnitude. Here, we remark that the kink of each ReLU neuron, i.e., the point where the output of ReLU is exactly zero, completely aligns with one of the input data points, which is consistent with the alignment behavior observed in Maennel et al. (2018). We also note that even though the weights and biases might take quite different values for each neuron as illustrated in Figure 1c, their activation points (or kinks) correspond to the data samples. The same analysis and conclusions also apply to binary classification scenarios with hinge loss as illustrated in Figure 1d. In this case, the resulting optimal networks are specific piecewise linear functions with kinks only at data points that determine the decision boundaries as the zero crossings. Based on these observations, the central questions we address in this paper are: *Why are over-parameterized ReLU NNs fitting a linear spline interpolation for one dimensional datasets? Is there a general mechanism inducing simple solutions in higher dimensions?* In the sequel, we show that these questions can be addressed by our convex analytic framework based on *convex geometry* and *duality*.

Here, we show that the optimal ReLU network fits a linear spline interpolation whose kinks at the input data points because the convex approximation[1] of a data point a_i given by

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

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 optimal hidden neuron activation thresholds for one dimensional ReLU networks. Similar characterizations also extend to the hidden neurons in multivariate cases as detailed below.

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 $\mathbf{a}_1, \ldots, \mathbf{a}_n \in^d$, the hidden neurons $\mathbf{u}_1, \ldots, \mathbf{u}_m \in^d$ 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 $k \in [n]$.

Notation: Matrices and vectors are denoted as uppercase and lowercase bold letters, respectively. $\mathbf{I}_k$ denotes the identity matrix of the size k. We use $(x)_+ = \max\{x, 0\}$ for the ReLU activation function. Furthermore, the set of integers from 1 to n are denoted as $[n]$

1. Here a_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 a_i as a convex mixture of the input data points in $\mathcal{S}\backslash\{i\}$.

and the notation $\mathbf{e}_j$ is used for the j^{th} ordinary basis vector. We also use $\mathcal{B}_2$ to denote the ℓ_2 unit ball in $\mathbb{R}^d$, i.e., $\mathcal{B}_2 = \{\mathbf{u} \in \mathbb{R}^d \mid \|\mathbf{u}\|_2 \leq 1\}$.

1.4 Organization of the Paper

The organization of this paper is as follows. In Section 2, we first describe the problem setting with the required preliminary concepts and then define notions of spike-free matrices and extreme points. Based on the definitions in Section 2, we then state our main results using convex duality in Section 3, where we also analyze some special cases, e.g., rank-one/whitened data, and introduce a training algorithm to globally optimize two-layer ReLU networks. We extend these results to various cases with regularization, vector outputs, arbitrary convex loss functions in Section 4. Here, we also provide closed-form solutions and/or equivalent convex optimization formulations for regularized ReLU network training problems. In Section 5, we briefly review the recently introduced NTK characterization and analytically compare it with our exact characterization. Then, Section 6 follows with numerical experiments on both synthetic and real benchmark datasets to verify our analysis in the previous sections. Finally, we conclude the paper with some remarks and future research directions in Section 7.

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 j^{th} hidden neuron, and $w_j \in \mathbb{R}$ is the corresponding output layer weight. 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]

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

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

where β is a regularization parameter. We define the overall parameter space Θ for (1) as $\theta \in \Theta = \{(\mathbf{U}, \mathbf{b}, \mathbf{w}, m) \,|\, \mathbf{U} \in \mathbb{R}^{d \times m}, \mathbf{b} \in \mathbb{R}^m, \mathbf{w} \in \mathbb{R}^m, m \in \mathbb{Z}_+\}$. Based on our observations in Figure 1a and the results in Savarese et al. (2019); Chizat and Bach (2018); Neyshabur

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 Maennel et al. (2018).
3. We defer the discussion of the more general vector output case to Section 4.5.

et al. (2014); Parhi and Nowak (2019), we first focus on a minimum norm variant of (2)[4]. 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.

Lemma 1 [5] *The following two optimization problems are equivalent:*

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

Lemma 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 1 and 2, we can express (3) as

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

However, both (2) and (4) are quite challenging optimization problems due to the optimization over hidden neurons and the ReLU activation. In particular, depending on the properties of $\mathbf{A}$, e.g., singular values, rank, and dimensions, the landscape of the non-convex objective in (2) can be quite complex.

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

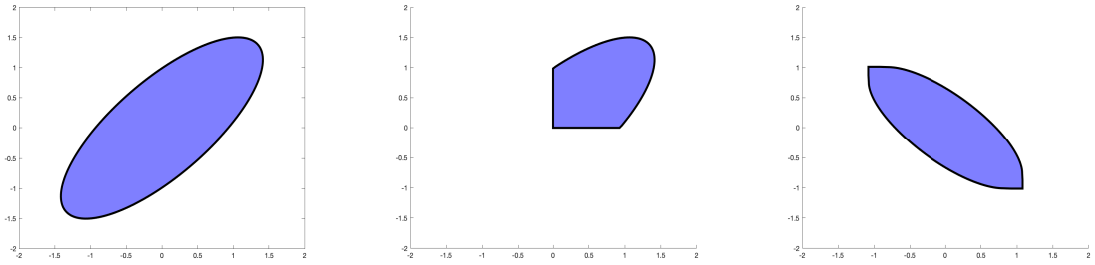
$$\min_{\mathbf{u}_1, w_1} \left\| w_1 (\mathbf{A}\mathbf{u}_1)_+ - \mathbf{y} \right\|_2^2 \text{ s.t. } \|\mathbf{u}_1\|_2 \leq 1. \tag{5}$$

The solution of (5) is completely determined by the set $\mathcal{Q}_\mathbf{A} = \{(\mathbf{A}\mathbf{u})_+ | \mathbf{u} \in \mathbb{R}^d, \|\mathbf{u}\|_2 \leq 1\}$. 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.

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. We note that the aforementioned set of matrices are, in fact, a more general class.

4. This corresponds to a weak form of regularization as $\beta \to 0$ in (2).
5. Proofs are presented in Appendix 11.



(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. Extreme points (spikes) induce the linear spline interpolation behavior in Figures 1b and 1d as predicted by our theory (see Lemma 7). The set shown in the middle figure acts as a regularizer analogous to a non-convex atomic norm.

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. 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 $\mathbf{u} = \mathbf{u}_1 w_1$

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

The following lemma provides a characterization of spike-free matrices

Lemma 3 *A matrix $\mathbf{A}$ is spike-free if and only if the following condition holds*

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

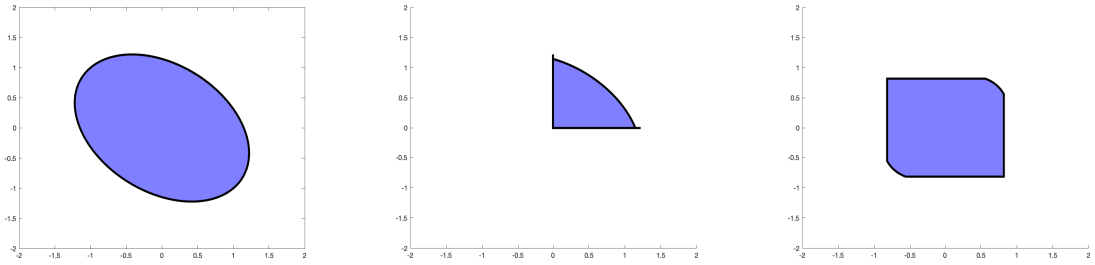
Alternatively, a matrix $\mathbf{A}$ is spike free if and only if it holds that

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

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

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

We note that the condition in (7) bears a close resemblance to the irrepresentability conditions in Lasso support recovery (see e.g. Zhao and Yu (2006)). 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

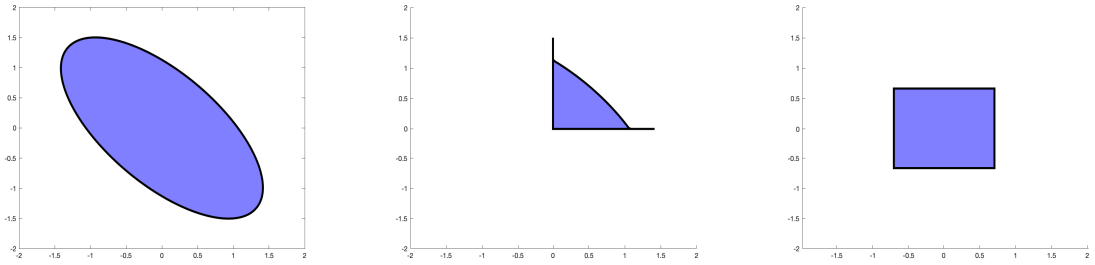


(a) Ellipsoidal set: $\{\mathbf{A}\mathbf{u} \,|\, \mathbf{u} \in \mathbb{R}^d, \|\mathbf{u}\|_2 \leq 1\}$

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



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

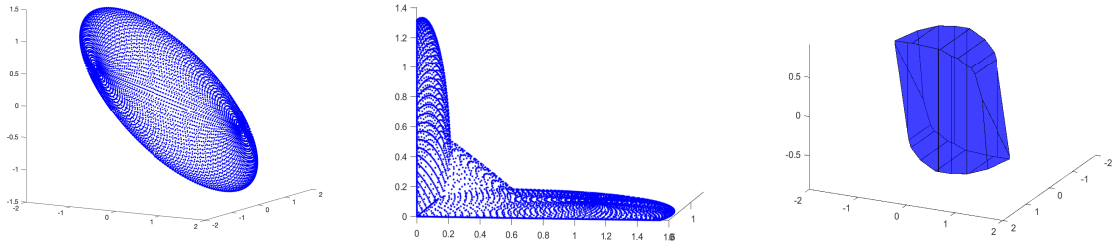
(b) Rectified ellipsoidal set $\mathcal{Q}_{\mathbf{A}}$: $\left\{\left(\mathbf{A}\mathbf{u}\right)_+ |\mathbf{u} \in \mathbb{R}^d, \|\mathbf{u}\|_2 \leq 1\right\}$

(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 4: Two dimensional illustration of a the rectified ellipsoid that is not spike-free and its polar set. Note that the polar set is polyhedral since the convex hull of the rectified ellipsoid is polyhedral.

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 $\tilde{\mathbf{A}}\tilde{\mathbf{A}}^T = \mathbf{I}_n$. Below, we show whitened data matrices are in fact spike-free. Furthermore, rank-one data matrices with positive left singular vectors are also spike-free as detailed below.



(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 rectified ellipsoid is the union of lower-dimensional ellipsoids its polar set exhibits a combination of smooth and non-smooth faces.

Lemma 4 *Let* $\mathbf{A}$ *be a whitened data matrix with* $n\leq d$ *that satisfies* $\mathbf{A}\mathbf{A}^T=\mathbf{I}_n$*. Then, it holds 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 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.*

Although the examples presented here for spike-free data matrices are whitened, one dimensional or rank-one, we believe that the set of spike-free data is far more generic and common. In addition, whitening is a common technique used by deep learning practitioners, e.g., ZCA whitening in image classification is used to improve the validation accuracy of the system as empirically shown in Coates and Ng (2012). Furthermore, recent work has shown that whitening improves the performance of the state-of-the-art architectures, e.g., ResNets, on benchmark datasets such as CIFAR-10, CIFAR-100, and ImageNet Huang et al. (2018). Therefore, we believe that theoretical results on spike-free matrices are valuable for practitioners.

2.3 Polar convex duality

It can be shown that the dual of the problem (4) is given by[6]

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

where $\mathcal{Q}_{\mathbf{A}}^\circ$ is the polar set (Rockafellar, 1970) of $\mathcal{Q}_{\mathbf{A}}$ defined as $\mathcal{Q}_{\mathbf{A}}^\circ=\{\mathbf{v}|\mathbf{v}^T\mathbf{u}\leq 1,\,\forall\mathbf{u}\in\mathcal{Q}_{\mathbf{A}}\}$.

6. We refer the reader to Appendix 12 for the proof.

2.4 Extreme points

A point is called an extreme point of a convex set $\mathcal{C}$ if it does not lie between any two distinct points of $\mathcal{C}$. More precisely, extreme points of a convex set $\mathcal{C}$ is defined as the set of points $\mathbf{v} \in \mathcal{C}$ such that if $\mathbf{v} = \frac{1}{2}\mathbf{v}_1 + \frac{1}{2}\mathbf{v}_2$, for some $\mathbf{v}_1, \mathbf{v}_2 \in \mathcal{C}$, then $\mathbf{v} = \mathbf{v}_1 = \mathbf{v}_2$. Let us also define the support function map

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

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 $\mathbf{v}$. In addition, $\sigma_{\mathcal{Q}_\mathbf{A}}(\mathbf{v})$ is not a singleton in general, but an exposed set. We also remark that the endpoints of the spikes in Figure 3 and 4 are the extreme points in the ordinary basis 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 points along an arbitrary direction.

Lemma 6 *In a one dimensional dataset ($d = 1$), 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 $i \in [n]$.*

The above lemma shows that the extreme point of the set $\mathcal{Q}_\mathbf{A}$ along an arbitrary direction $\mathbf{v}$ yields a set of hidden neurons and biases that take values of ± 1 and $\pm a_i$, respectively, for an arbitrary index $i \in [n]$. Therefore, the kink of each ReLU activation at the extreme points corresponds to one of the input data samples, i.e., $a_i u + b = 0$ for an arbitrary index $i \in [n]$. Moreover, in the next section (Theorem 1 and other results) we prove the optimality of these extreme points. Therefore, combined with Theorem 1, the above result proves that the optimal network outputs the linear spline interpolation for the input data.

We now generalize the result to higher dimensions by including the extreme points in the span of the ordinary basis vectors. These will improve our spike-free relaxation as a first order correction. In particular, for the spike-free relaxation, we represent the output of ReLU activation as $\big(\mathbf{Au}\big)_+ = \mathbf{Au}$ with the constraint $\mathbf{Au} \succcurlyeq \mathbf{0}$. However, this representation is restrictive since it doesn't allow any non-negative preactivation $\mathbf{Au}$. In order to further improve this relaxation, we also include extreme points in the ordinary basis directions $\{\mathbf{e}_i\}_{i=1}^n$. Therefore, instead of restricting preactivations to be all nonnegative, we also allow preactivations that have one positive entry. 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 7 *Extreme point in the span of each ordinary basis direction $\mathbf{e}_i$ is given by*

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

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

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

Lemma 7 shows that extreme points of $\mathcal{Q}_{\mathbf{A}}$ are given by a convex mixture approximation of the training samples: the hidden neurons are the residuals of the approximation and the corresponding bias values is the negative inner product between the hidden neurons and a training sample, which places a kink at the training sample. Our next result characterizes extreme points along arbitrary directions for the general case.

Lemma 8 *For any* $\boldsymbol{\alpha} \in \mathbb{R}^n$*, the extreme point along the direction of* $\boldsymbol{\alpha}$ *can be found by*

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

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

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

Lemma 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 1 (A representer theorem for optimal neurons) *Lemma 8 implies that each extreme point along the direction* $\boldsymbol{\alpha}$ *can be written in the following compact form*

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

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

Remark 1 *An interpretation of the extreme points provided above is an auto-encoder of the training data: the optimal neurons are convex mixture approximations of subsets of training samples in terms of 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 for the non-convex NN training objective given in (4).

Theorem 1 *The dual of the problem in* (4) *is given by*

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

and we have $P^* \geq D^*$. For finite width NNs, there exists an optimal network width m^* that is upperbounded as $m^* \leq n+1$ such that strong duality holds, 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.*

Remark 2 *Over-parameterization has been extensively studied in the literature and has shown to be one of the key factors for the remarkable generalization performance of deep neural networks Du et al. (2018); Li and Liang (2018); Du and Lee (2018); Arora et al. (2018); Brutzkus et al. (2017); Neyshabur et al. (2018); Jacot et al. (2018). However, most of these studies require an extreme over-parameterization level, e.g., (Du et al, 2018) Du et al. (2018) requires $m = \mathcal{O}(n^6)$, or even an infinite-width as in Jacot et al. (2018), which is far from empirical observations and therefore fail to explain the success of neural networks in practice. However, the result in Theorem 1 require m^* neurons, where $m^* \leq n+1$. Even the upper-bound $n+1$ on m^* is significantly less over-parameterized than the previous theoretical studies. Therefore, we claim that our analysis requires a moderate amount of overparameterization and aligns with practical settings.*

Remark 3 *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. Guruswami and Raghavendra (2009); Bach (2017)).*

Based on the dual form and the optimality condition in Theorem 1, we can characterize the optimal neurons as the extreme points of a certain set.

Corollary 2 *Theorem 1 implies that the optimal neuron weights are extreme points which solve the following optimization problem*

$$\operatorname*{argmax}_{\mathbf{u}\in\mathcal{B}_2} \left|\mathbf{v}^{*T}\left(\mathbf{A}\mathbf{u}\right)_+\right|.$$

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 first provide a theoretical analysis for the duality gap of finite width NNs and then prove strong duality under certain technical conditions.

3.1.1 Duality for finite width neural networks

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

Theorem 2 *Suppose that the optimization problem* (4) *is feasible, i.e., there exists a set* θ *such that* $f_\theta(\mathbf{A}) = \mathbf{y}$*, then weak duality holds for* (4)*.*

We now prove that strong duality holds for any feasible finite width NN, in which the number of neurons exceeds a critical number less than $n+1$.

Theorem 3 *Let* $\{\mathbf{A}, \mathbf{y}\}$ *be a dataset such that the optimization problem* (4) *is feasible and the width exceeds a critical threshold, i.e.,* $m \geq m^*$*, where* m^* *is the number of constraints active in the dual problem* (12) *that obeys* $m^* \leq n+1$*. Then, strong duality holds for* (4)*.*

Since strong duality holds for finite width NNs as proved in Theorem 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 2.

In the sequel, we first show that we can explicitly characterize the set of extreme points for some specific practically relevant problems. We then prove that strong duality holds for these problems.

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 6.

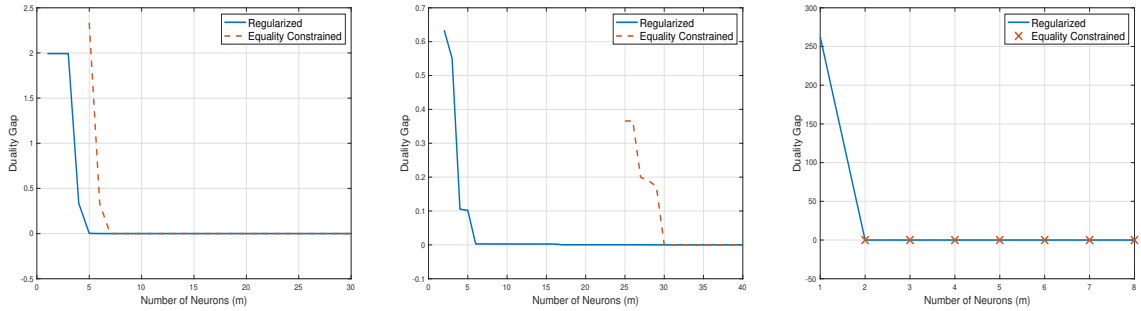
Corollary 3 *Let* $\{a_i\}_{i=1}^n$ *be a one dimensional training set i.e.,* $a_i \in \mathbb{R}$*,* $\forall i \in [n]$*. Then, a set of solutions to* (4) *that achieve the optimal value are extreme points, and therefore satisfy* $\{(u_i, b_i)\}_{i=1}^m$*, where* $u_i = \pm 1, b_i = -sign(u_i)a_i$*.*

Corollary 4 *For problems involving one dimensional training data, strong duality holds as a result of Corollary 3 and Theorem 3 when* $m \geq n+1$*.*

Corollary 4 implies that we can globally optimize (1) using a subset of finite number of solutions in Corollary 3. In Figure 6a, we perform a numerical experiment on the dataset plotted in Figure 1b. Since strong duality holds for finite width networks with at most $n+1$ neurons as proven in Theorem 1, in Figure 6a, the duality gap vanishes when we reach a certain m value using the parameters formulated in Corollary 3. Notice that this result also validates Theorem 3.

Besides, Corollary 3 proves that the optimal function output is the linear spline interpolation for the input data, where the kinks of ReLU activations occur at one of the data points. However, in the following, we prove that this set of solutions is not unique so that there might exist other optimal solutions to (4) with different function outputs.

Proposition 1 *The solution provided in Corollary 3 is not unique in general. Let us denote the set of active samples for an arbitrary neuron with the parameters* $(\mathbf{u}, b)$ *as* $\mathcal{S} = \{i | a_i u + b \geq 0\}$ *and its complementary* $\mathcal{S}^c = \{j | a_j u + b < 0\} = [n]/\mathcal{S}$*. Then, whenever* $\sum_{i \in \mathcal{S}} v_i = 0$*,*



(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 in (4) and the regularized case in (16). To solve the primal and dual problems, we use CVX Grant and Boyd (2014) with the SDPT3 solver Tütüncü et al. (2001).

the value of the bias does not change the objective in the dual constraint. Thus, all the bias values in the range $[\max_{i\in\mathcal{S}}(-a_i),\ \min_{j\in\mathcal{S}^c}(-a_j)]$ *are optimal. In such cases, there are multiple optimal solutions for the training problem. Remarkably, our duality framework enables the construction of all optimal solutions. In Appendix 11, we present an analytic expressions for a counter-example where an optimal solution is not in this form, i.e., not a piecewise linear spline, and illustrate the corresponding function fits in Figure 7.*

We remark that Savarese et al. (2019) also proved the optimality of the linear spline interpolation as the network output function for one dimensional data and scalar output networks. Moreover, they empirically observed the non-uniqueness of the optimal solutions however did not give any theoretical justifications for this observation. On the contrary, in Proposition 1, we completely characterize the set of optimal solutions via convex duality and prove that the linear spline interpolation is not the only optimal solution. In other words, there might exist other optimal solutions with different functional forms. Based on other characterization for the set of optimal solutions, we also provide an empirical evidence for non-uniqueness in Figure 7. Here, we particularly depict three different optimal function fits that are very similar to the linear spline interpolation with the kinks at the input data points except a kink in the range $(0, 1)$. Furthermore, since we utilize a more generic approach based on convex duality, our analysis is valid for rank-one data and vector output networks as detailed in the next section and Section 4.5, respectively.

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 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, 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* $s_i = \pm 1, \forall i \in [m]$.

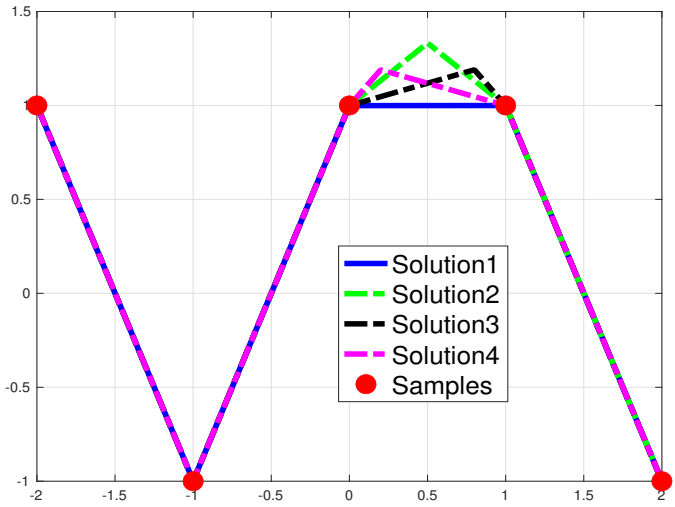


Figure 7: An example verifying our non-uniqueness characterization in Proposition 1. Here, we plot four different solutions that analytically achieve the optimal ℓ_2^2 regularized training cost but produce different predictions. It is interesting to note that the minimum norm criteria is not sufficient to uniquely determine the NN model. The details (including the exact analytical expressions) are presented in Section 11.

Corollary 6 *As a result of Theorem 3 and Corollary 5, strong duality holds for problems involving rank-one data matrices.*

Corollary 5 and 6 indicate that we can globally optimize regularized NNs using a subset of the extreme points in Corollary 5. We also present a numerical example in Figure 6b to confirm the theoretical prediction of Corollary 6.

3.4 Solutions to spike-free problems

Here, we show that as a direct consequence of our analysis above, problems involving spike-free data matrices can be equivalently stated as a convex optimization problem. The next result formally presents the convex equivalent problem and further proves that this problem can be globally optimized in polynomial-time with respect to all the problem parameters n, d, and m.

Theorem 4 [7] *Let* $\mathbf{A}$ *be a spike-free data matrix. Then, the non-convex training problem* (4) *can be equivalently formulated as the following convex optimization problem*

$$\min_{\mathbf{w}_1,\mathbf{w}_2} \|\mathbf{w}_1\|_2 + \|\mathbf{w}_2\|_2 \ \ s.t. \ \ \mathbf{A}(\mathbf{w}_1 - \mathbf{w}_2) = \mathbf{y}, \ \ \mathbf{A}\mathbf{w}_1 \succcurlyeq \mathbf{0}, \ \ \mathbf{A}\mathbf{w}_2 \succcurlyeq \mathbf{0},$$

which can be globally optimized by a standard convex optimization solver in $\mathcal{O}(d^3)$.

7. Proof and extensions are presented in Section 4.4.

Theorem 4 proves that the regularized training problem for spike-free data matrix can be equivalently cast as a convex optimization problem with two neurons. More importantly, this convex problem can be globally optimized by standard convex optimization solvers, e.g., interior-point methods, with a polynomial-time complexity in terms of the number of data samples n and the feature dimension d.

3.5 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. Candes and Tao (2005); Donoho (2006); Fung and Mangasarian (2011)). 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 9 *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*

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

Lemma 10 *We have ℓ_1-ℓ_0 equivalence, i.e., the optimal solutions of* (13) *and* (4) *coincide if the following condition holds*

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

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

Corollary 7 *As a result of Theorem 3 and Corollary 10, strong duality holds for problems involving whitened data matrices.*

Lemma 10 and Corollary 7 prove that (1) can be globally optimized using the two extreme points in Lemma 9. Therefore, we achieve an equivalence between ℓ_1-ℓ_0 problems. Moreover, this result is also consistent with Theorem 4 as its special case, i.e., we have full row-rank spike-free data matrix. We also validate Corollary 7 via a numerical experiment in Figure 6c.

3.6 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

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

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 (Frank and Wolfe, 1956) in order to approximate the solution iteratively. In the following, we show how to relax the problem using our spike-free relaxation

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

where we relax the set $\{(\mathbf{A}\mathbf{u})_+|\mathbf{u}\in\mathbb{R}^d, \|\mathbf{u}\|_2\leq 1\}$ as $\{\mathbf{A}\mathbf{u}|\mathbf{u}\in\mathbb{R}^d, \|\mathbf{u}\|_2\leq 1\}\cap\mathbb{R}^n_+$. 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 neurons 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 $\mathbf{U}$ and then repeat this procedure till we find an optimal solution. We also provide the full algorithm in Algorithm 1[8].

The major advantage of our cutting-plane algorithm (compared to the non-convex methods such as Frank-Wolfe in Bach (2017)) is that it can be solved via convex optimization. More specifically, in Algorithm 1, all the problems we need to solve are convex and therefore can be globally and efficiently optimized by standard convex solvers without requiring any exhaustive search to tune hyperparameters, e.g., learning rate and initialization, or heuristics such as dropout. Moreover, since the algorithm incrementally inserts neurons to the hidden layer, there is no need to carefully tune the hidden layer width or use unnecessarily wide architectures to fit the training data. We next prove the convergence of Algorithm 1 for spike-free data.

Proposition 2 *When* $\mathbf{A}$ *is spike-free as defined in Lemma 3, the cutting plane based training method globally optimizes* (12)*.*

The following theorem shows that spike-free cases are in fact practically relevant. Particularly, random high dimensional i.i.d. Gaussian matrices asymptotically satisfy the spike-free condition as detailed below.

Theorem 5 *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*

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

We now consider improving the basic spike-free relaxation by including the extreme points along the ordinary basis vectors $\{\mathbf{e}_i\}_{i=1}^n$, which is detailed in Lemma 7. The next result characterizes the cases where employing only these extreme points are sufficient to fit the training data.

8. We also provide the cutting plane method for NNs with a bias term in Appendix 9.

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 $\hat{\mathbf{u}}_2$:

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

4: $\mathbf{U}^* \leftarrow [\mathbf{U}^* \ \hat{\mathbf{u}}_1 \ \hat{\mathbf{u}}_2]$
5: Find $\mathbf{v}$ by solving the dual problem:

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

6: Check the existence of a violating neuron:

$$\max_{\mathbf{u}:\mathbf{A}\mathbf{u}\succcurlyeq\mathbf{0},\|\mathbf{u}\|_2\leq 1} \mathbf{v}^T\mathbf{A}\mathbf{u} \overset{?}{\geq} 1 \qquad \min_{\mathbf{u}:\mathbf{A}\mathbf{u}\succcurlyeq\mathbf{0},\|\mathbf{u}\|_2\leq 1} \mathbf{v}^T\mathbf{A}\mathbf{u} \overset{?}{\leq} -1,$$

7: **end while**
8: Solve the training problem using $\mathbf{U}^*$:

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \|\mathbf{w}\|_1 \text{ s.t. } \left(\mathbf{A}\mathbf{U}^*\right)_+\mathbf{w} = \mathbf{y}$$

9: Return $\theta = (\mathbf{U}^*, \mathbf{w}^*)$

Theorem 6 *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 7 achieve zero training error.*

This shows that the extreme points in Lemma 7 enable the network architecture to achieve zero training error and therefore completely characterize the training data geometry when each data sample is a vertex of the convex hull of all the samples.

Our next result shows that the above claim, i.e., " each data sample is a vertex of the convex hull of all the samples", is likely to hold for high dimensional random matrices.

Theorem 7 *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.*

4. Regularized ReLU Networks and Convex Optimization for Spike-Free Matrices

In this section, we present extensions of our approach to regularized networks, arbitrary convex loss functions, and vector outputs. More importantly, as a corollary of our analysis in the previous section, we provide exact convex formulations for problems with spike-free

data matrices and show that convex equivalent formulations can be globally optimized in polynomial-time by a standard convex solver.

4.1 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 8 *Optimal hidden neurons* $\mathbf{U}$ *for the following regularized problem*

$$\min_{\theta\in\Theta} \frac{1}{2}\|(\mathbf{A}\mathbf{U})_+\mathbf{w}-\mathbf{y}\|_2^2+\frac{\beta}{2}\left(\|\mathbf{w}\|_2^2+\|\mathbf{U}\|_F^2\right)=\min_{\theta\in\Theta} \frac{1}{2}\|(\mathbf{A}\mathbf{U})_+\mathbf{w}-\mathbf{y}\|_2^2+\beta\|\mathbf{w}\|_1 \ s.t. \ \|\mathbf{u}_j\|_2\leq 1, \forall j, \tag{16}$$

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. Here* $\mathcal{Q}_{\mathbf{A}}^{\circ}$ *is the convex polar of the rectified ellipsoid* $\mathcal{Q}_{\mathbf{A}}$*.*

Remark 4 *Based on Theorem 8, we note that all the weak and strong duality results in Theorem 2 and 3 hold for regularized networks. Therefore, Corollary 4, 6 and 7 also apply to regularized networks. Furthermore, the numerical results in Figure 6 confirm this claim.*

Corollary 8 *Remark 4 also implies that whenever the set of extreme points can be explicitly characterized, 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*

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

or the corresponding regularized version

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

Next, we provide the closed-form formulations for the optimal solutions to the regularized training problem (16) as in Lemma 9.

Theorem 9 *Suppose* $\mathbf{A}$ *is whitened such that* $\mathbf{A}\mathbf{A}^T=\mathbf{I}_n$*, then an optimal solution set for* (16) *can be formulated as follows*

$$(\mathbf{U}^*,\mathbf{w}^*)=\begin{cases}\left(\left[\frac{\mathbf{A}^\dagger\left(\mathbf{y}\right)_+}{\|\mathbf{A}^\dagger\left(\mathbf{y}\right)_+\|_2},\frac{\mathbf{A}^\dagger\left(-\mathbf{y}\right)_+}{\|\mathbf{A}^\dagger\left(-\mathbf{y}\right)_+\|_2}\right],\begin{bmatrix}\|\mathbf{A}^\dagger\left(\mathbf{y}\right)_+\|_2-\beta\\ -\|\mathbf{A}^\dagger\left(-\mathbf{y}\right)_+\|_2+\beta\end{bmatrix}\right) & \text{if } \beta\leq\|\left(\mathbf{y}\right)_+\|_2,\ \beta\leq\|\left(-\mathbf{y}\right)_+\|_2\\ \left(\frac{\mathbf{A}^\dagger\left(-\mathbf{y}\right)_+}{\|\mathbf{A}^\dagger\left(-\mathbf{y}\right)_+\|_2},-\|\mathbf{A}^\dagger\left(-\mathbf{y}\right)_+\|_2+\beta\right) & \text{if } \beta>\|\left(\mathbf{y}\right)_+\|_2,\ \beta\leq\|\left(-\mathbf{y}\right)_+\|_2\\ \left(\frac{\mathbf{A}^\dagger\left(\mathbf{y}\right)_+}{\|\mathbf{A}^\dagger\left(\mathbf{y}\right)_+\|_2},\|\mathbf{A}^\dagger\left(\mathbf{y}\right)_+\|_2-\beta\right) & \text{if } \beta\leq\|\left(\mathbf{y}\right)_+\|_2,\ \beta>\|\left(-\mathbf{y}\right)_+\|_2\\ (\mathbf{0},0) & \text{if } \beta>\|\left(\mathbf{y}\right)_+\|_2,\ \beta>\|\left(-\mathbf{y}\right)_+\|_2\end{cases}.$$

We note that Theorem 9 exactly characterizes how the regularization parameter β controls the number of neurons and changes the analytical form of the optimal network parameters. Therefore, for whitened data matrices, there is no need to train a network via backpropagation or other numerical methods, instead, one can directly utilize the closed-form solutions in Theorem 9.

4.2 Two-layer ReLU networks with hinge loss

Now we consider classification problems with the label vector $\mathbf{y} \in \{+1, -1\}^n$ and hinge loss.

Theorem 10 *An optimal* $\mathbf{U}$ *for the binary classification task with the hinge loss given by*

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

can be found through the following dual

$$\max_{\mathbf{v}} \mathbf{v}^T \mathbf{y} \ \ s.t. \ \ 0 \leq y_i v_i \leq 1 \ \forall i \in [n], \mathbf{v} \in \beta \mathcal{Q}_{\mathbf{A}}^{\circ}, -\mathbf{v} \in \beta \mathcal{Q}_{\mathbf{A}}^{\circ} \, .$$

Theorem 10 proves that since strong duality holds for two-layer 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).

Corollary 9 *Theorem 10 implies that the optimal neuron weights are extreme points which solve the following optimization problem*

$$\underset{\mathbf{u} \in \mathcal{B}_2}{\operatorname{argmax}} \left| \mathbf{v}^{*T} \left(\mathbf{A}\mathbf{u} \right)_+ \right| ,$$

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

Consequently, in the one dimensional case, the optimal neuron weights are given by the extreme points. Therefore the optimal network network output is given by the piecewise linear function

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

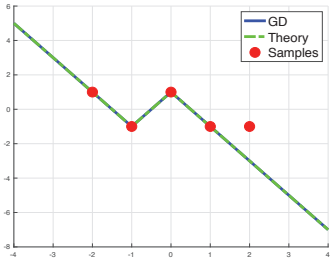
for some output weights $w_1, \ldots, w_m$ where $u_j = \pm 1$ and $b_j = \mp a_j$ for some j.

This explains Figure 1d, 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.

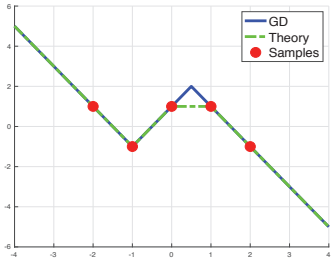
It is easy to see that this is equivalent to applying the kernel map $\kappa(a, a_j) = (a - a_j)_+$, forming the corresponding kernel matrix

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

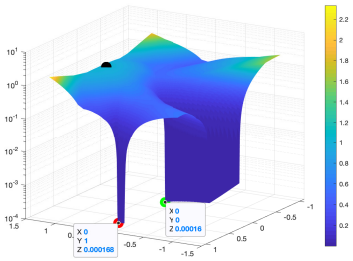
and solving minimum ℓ_1-norm SVM on the kernelized data matrix. The same steps can also be applied to a rank-one dataset as presented in the following.



(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 $L_{gd} = 1.679 \times 10^{-4}$.



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

Figure 8: Binary classification using hinge loss, where we apply GD and our approach in Theorem 11. Here, we denote the objective value in (18) as L_t and L_{gd} for our theoretical approach (Theorem 11) and GD, respectively. We also note that optimal solution might not be unique as shown in Proposition 1 and Figure 7, which explains why GD converges to a solution with a kink in the middle of two data points in (b). 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 10 *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* $s_i = \pm 1, \forall i \in [m]$.

Theorem 11 *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 of Corollary 10 and Theorem 11 directly follows from the proof of Corollary 5.

We also verify Theorem 11 using a one dimensional dataset in Figure 8. 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 need to create any kink as in Figure 8a. However, we also note that optimal solution is not unique and there might exist other optimal solutions as shown in Proposition 1 and Figure 7. This is exactly what we observe in in Figure 8b, where GD try to converge to a solution with a kink in the middle of two data points unlike our approach. In Figure 8c, we also provide a visualization of the loss landscape for this case.

Theorem 12 *Suppose that* $\mathbf{A}$ *is whitened such that* $\mathbf{A}\mathbf{A}^T = \mathbf{I}_n$. *Then an optimal solution to the problem in* (18) *is given by*

$$(\mathbf{u}_1^*, w_1^*) = \begin{cases} \left(\frac{\mathbf{A}^\dagger \left(\mathbf{y}\right)_+}{\sqrt{n_+}}, w_+ \right) & \text{if } \beta = \sqrt{n_+} \\ \left(\frac{\mathbf{A}^\dagger \left(\mathbf{y}\right)_+}{\sqrt{n_+}}, \sqrt{n_+} \right) & \text{if } \beta < \sqrt{n_+} \\ (\mathbf{0}, 0) & \text{if } \beta > \sqrt{n_+} \end{cases} \quad (\mathbf{u}_2^*, w_2^*) = \begin{cases} \left(\frac{\mathbf{A}^\dagger \left(-\mathbf{y}\right)_+}{\sqrt{n_-}}, w_- \right) & \text{if } \beta = \sqrt{n_-} \\ \left(\frac{\mathbf{A}^\dagger \left(-\mathbf{y}\right)_+}{\sqrt{n_-}}, -\sqrt{n_-} \right) & \text{if } \beta < \sqrt{n_-} \\ (\mathbf{0}, 0) & \text{if } \beta > \sqrt{n_-} \end{cases},$$

where $w_+ \in [0, \sqrt{n_+}]$, $w_- \in [-\sqrt{n_-}, 0]$, n_+, *and* n_- *are the number of samples with positive and negative labels, respectively.*

Theorem 12 shows that the well-known problem achieving minimum ℓ_2-norm parameters maximizing the margin between two classes can be solved in closed-form for some generic settings such as problems involving a spike-free and/or whitened data matrix. As in the squared loss case, we observe that the weight decay parameter β directly controls the sparsity of the optimal solution.

Interpretation of the hidden neurons as Fisher's Linear Discriminant vectors:
According to Theorem 12, it is interesting to note that for generic full column rank matrices and binary labels $\mathbf{y} \in \{+1, -1\}^n$, the expression for the first hidden neuron equals $\mathbf{u}_1^* = \mathbf{A}^\dagger(\mathbf{y})_+ = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T(\mathbf{y})_+ = \hat{\mathbf{\Sigma}}^{-1}\hat{\boldsymbol{\mu}}_1$ where $\hat{\mathbf{\Sigma}} := \frac{1}{n}\mathbf{A}^T\mathbf{A}$ is the empirical covariance matrix and $\boldsymbol{\mu}_1 = \frac{1}{n}\mathbf{A}^T(\mathbf{y})_+ = \frac{1}{n}\sum_{i:y_i>0}\mathbf{a}_i$ is the empirical mean of the samples in the first class. In fact, this formula is identical to Fisher's Linear Discriminant for binary classification. For whitened data, note that $\hat{\mathbf{\Sigma}}$ is a multiple of the identity matrix. This shows that there are two optimal hidden neurons for two individual classes corresponding to the $\{+1, -1\}$ labels. Furthermore, a particular hidden neuron is only active when the weight decay parameter β is less than the square root of the number of samples in the corresponding class, i.e., $\beta < \sqrt{n_-}$ or $\beta < \sqrt{n_+}$. In theorem 15, we provide a closed-form expression for the multi-class case where the number of hidden neurons can be as large as the number of classes and is controlled by the magnitude of β.

4.3 Two-layer ReLU networks with general loss functions

Now we consider the scalar output two-layer ReLU networks with a generic loss

$$\min_{\theta\in\Theta} \ell((\mathbf{A}\mathbf{U})_+\mathbf{w}, \mathbf{y}) + \frac{\beta}{2}(\|\mathbf{w}\|_2^2 + \|\mathbf{U}\|_F^2) = \min_{\theta\in\Theta} \ell((\mathbf{A}\mathbf{U})_+\mathbf{w}, \mathbf{y}) + \beta\|\mathbf{w}\|_1 \text{ s.t. } \|\mathbf{u}_j\|_2 \leq 1, \forall j, \tag{19}$$

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

Theorem 13 *The dual of* (19) *is given by*

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

where ℓ^* *is the Fenchel conjugate function defined as (Boyd and Vandenberghe, 2004)*

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

Theorem 13 proves that our extreme point characterization in Lemma 6 applies to arbitrary convex loss functions. 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 the loss function.

4.4 Polynomial-time convex optimization of problems with spike-free matrices

Here, we show that two-layer ReLU network training problems involving spike-free data matrices can be equivalently stated as convex optimization problems. More importantly, we prove that the equivalent form can be globally optimized by standard convex optimization solvers with polynomial complexity in terms of the number of data samples n and the data dimension d.

We start by restating the two-layer training problem with arbitrary convex loss functions as follows

$$\min_{\theta\in\Theta} \ell((\mathbf{AU})_+\mathbf{w},\mathbf{y}) + \beta\|\mathbf{w}\|_1 \text{ s.t. } \|\mathbf{u}_j\|_2 \leq 1, \forall j. \tag{20}$$

Then, by Theorem 1 and 13, we have the following dual problem with respect to the output layer weights $\mathbf{w}$

$$\max_{\mathbf{v}} -\ell^*(\mathbf{v}) \text{ s.t. } \max_{\substack{\mathbf{u}\in\mathcal{B}_2 \\ \mathbf{Au}\succcurlyeq\mathbf{0}}} \left|\mathbf{v}^T\mathbf{Au}\right| \leq 1, \tag{21}$$

where we replace $(\mathbf{Au})_+$ with $\{\mathbf{Au} : \mathbf{Au} \succcurlyeq \mathbf{0}\}$ since $\mathbf{A}$ is spike-free. If we take the dual of (21) with respect to $\mathbf{v}$ one more time, i.e., also known as the bidual of (20), we obtain the following optimization problem

$$\min_{\mathbf{w}_1,\mathbf{w}_2} \ell(\mathbf{A}(\mathbf{w}_1-\mathbf{w}_2),\mathbf{y}) + \beta(\|\mathbf{w}_1\|_2 + \|\mathbf{w}_2\|_2) \text{ s.t. } \mathbf{Aw}_1 \succcurlyeq \mathbf{0},\ \mathbf{Aw}_2 \succcurlyeq \mathbf{0}. \tag{22}$$

Note that (22) is a convex optimization problem with $2d$ variables and $2n$ constraints. Therefore, standard interior-point solvers, e.g., CVX Grant and Boyd (2014) with the SDPT3 solver Tütüncü et al. (2001), to solve convex optimization problems in Section 4.4. can globally optimize (22) with the computational complexity $\mathcal{O}(d^3)$.

4.5 Extension to vector output neural networks

In this section, we first derive the dual form for vector output NNs and then describe the implementation of the cutting plane algorithm. For presentation simplicity, we consider the weakly regularized scenario with squared loss. However, all the derivations in this section can be extended to regularized problems with arbitrary convex loss functions as proven in the previous section.

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, the training problem is as follows

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

Lemma 11 *The following two optimization problems are equivalent:*

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

Using Lemma 11, we get the following equivalent form

$$\min_{\theta\in\Theta} \sum_{j=1}^{m} \|\mathbf{w}_j\|_2 \text{ s.t. } \left(\mathbf{AU}\right)_+\mathbf{W} = \mathbf{Y},\ \|\mathbf{u}_j\|_2 \leq 1,\ \forall j, \tag{23}$$

which has the following dual form

$$\max_{\mathbf{V}} \text{trace}(\mathbf{V}^T\mathbf{Y}) \text{ s.t. } \|\mathbf{V}^T(\mathbf{Au})_+\|_2 \leq 1,\ \mathbf{u} \in \mathcal{B}_2. \tag{24}$$

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

$$\hat{\mathbf{u}} = \underset{\mathbf{u}}{\text{argmax}} \|\mathbf{V}^T\mathbf{Au}\|_2 \text{ s.t. } \mathbf{Au} \succcurlyeq \mathbf{0}, \|\mathbf{u}\|_2 \leq 1 \tag{25}$$

Therefore, the hidden layer weights can be determined by solving the above optimization problem.

4.5.1 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 (24) can be obtained via the following maximization problem

$$\underset{u,b}{\text{argmax}} \|\mathbf{V}^T(\mathbf{a}u + b\mathbf{1})_+\|_2^2 \text{ s.t. } |u| = 1. \tag{26}$$

Using the same steps in Proof of Lemma 6, we can write (26) as follows

$$\underset{u,b}{\text{argmax}} \Big\| \sum_{i\in\mathcal{S}} \mathbf{v}_i(a_iu + b)\Big\|_2^2 \text{ s.t. } a_iu + b \geq 0, \forall i \in \mathcal{S}, a_ju + b \leq 0, \forall j \in \mathcal{S}^c, |u| = 1. \tag{27}$$

Notice that u can be either $+1$ or -1. Thus, we can solve the problem for each option and then pick the one with higher objective value. First assume that $u = +1$, then (27) becomes

$$\underset{b}{\text{argmax}} \Big\| \sum_{i\in\mathcal{S}} \mathbf{v}_i(a_i + b)\Big\|_2^2 \text{ s.t. } \max_{i\in\mathcal{S}} -a_i \leq b \leq \min_{j\in\mathcal{S}^c} -a_j. \tag{28}$$

Since

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

(28) is a convex function of b. Therefore, the optimal solution to (28) 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 11 *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* (23) *that achieve the optimal value satisfy $\{(u_i, b_i)\}_{i=1}^m$, where $u_i = \pm 1, b_i = -sign(u_i)a_i$.*

4.5.2 Solutions to rank-one problems

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

$$\begin{aligned}\underset{b,\mathbf{u}:\|\mathbf{u}\|_2=1}{\operatorname{argmax}} \left\| \mathbf{V}^T\big(\mathbf{A}\mathbf{u} + b\mathbf{1}\big)_+ \right\|_2^2 &= \underset{b,\mathbf{u}:\|\mathbf{u}\|_2=1}{\operatorname{argmax}} \left\| \mathbf{V}^T\big(\mathbf{c}\mathbf{a}^T\mathbf{u} + b\mathbf{1}\big)_+ \right\|_2^2 \\ &= \underset{b,\mathbf{u}:\|\mathbf{u}\|_2=1}{\operatorname{argmax}} \Big\| \sum_{i=1}^n \mathbf{v}_i\big(c_i\mathbf{a}^T\mathbf{u} + b\big)_+ \Big\|_2^2,\end{aligned}$$

which can be equivalently stated as

$$\begin{aligned}&\underset{b,\mathbf{u}:\|\mathbf{u}\|_2=1}{\operatorname{argmax}} \Big\| \sum_{i\in\mathcal{S}} \mathbf{v}_i c_i \Big\|_2^2 (\mathbf{a}^T\mathbf{u})^2 + 2b(\mathbf{a}^T\mathbf{u}) \Big(\sum_{i\in\mathcal{S}} \mathbf{v}_i c_i \Big)^T \sum_{i\in\mathcal{S}} \mathbf{v}_i + b^2 \Big\| \sum_{i\in\mathcal{S}} \mathbf{v}_i \Big\|_2^2 \\ &\text{s.t. } \begin{cases} c_i\mathbf{a}^T\mathbf{u} + b \geq 0, \forall i \in \mathcal{S} \\ c_j\mathbf{a}^T\mathbf{u} + b \leq 0, \forall j \in \mathcal{S}^c \end{cases},\end{aligned}$$

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 12 *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 (23) 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 $s_i = \pm 1, \forall i \in [m]$.*

Theorem 14 *For one dimensional and/or rank-one datasets, solving the following ℓ_2-norm convex optimization problem globally optimizes (23)*

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

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

Theorem 14 proves that for one dimensional and/or rank-one datasets, the regularized non-convex training problem in (23) can be equivalently stated as a convex optimization problem where the hidden neurons are fixed. Therefore, the training problem can be globally and efficiently optimized via standard convex optimization solvers.

4.5.3 Solutions to Whitened Problems

Here, we provide the closed-form formulations for the optimal solutions to the regularized training problem (23) as in Theorem 9.

Theorem 15 *Let $\{\mathbf{A}, \mathbf{Y}\}$ be a dataset such that $\mathbf{A}\mathbf{A}^T = \mathbf{I}_n$ and $\mathbf{Y}$ is one hot encoded label matrix, then a set of optimal solution $(\mathbf{U}^*, \mathbf{W}^*)$ to*

$$\min_{\theta \in \Theta} \frac{1}{2} \| \left(\mathbf{A}\mathbf{U}\right)_+ \mathbf{W} - \mathbf{Y} \|_F^2 + \frac{\beta}{2} \left(\|\mathbf{U}\|_F^2 + \|\mathbf{W}\|_F^2 \right)$$

can be formulated as follows

$$\left(\mathbf{u}_j^*, \mathbf{w}_j^*\right) = \begin{cases} \left(\frac{\mathbf{A}^\dagger \mathbf{y}_j}{\|\mathbf{A}^\dagger \mathbf{y_j}\|_2}, \left(\sqrt{n_j} - \beta\right) \mathbf{e}_j \right) & \text{if } \beta \leq \sqrt{n_j} \\ (\mathbf{0}, \mathbf{0}) & \text{otherwise} \end{cases}, \quad \forall j \in [o],$$

where n_j is the number samples in class j and $\mathbf{e}_j$ is the j^{th} ordinary basis vector.

The above result shows that there are at most o hidden neurons, which individually correspond to classifying a particular class. A hidden neuron is only active when the weight decay parameter β is less than the square root of the number of samples in the corresponding class. It is interesting to note that the form of the hidden neuron is identical to Fisher's Linear Discriminant for binary classification applied in a one-vs-all fashion.

4.5.4 Convex optimization for spike-free problems

We now analyze the case when the data matrix is spike-free. Following the approach in Section 4.4, we first state the dual problem as follows

$$\max_{\mathbf{V}} \operatorname{trace}(\mathbf{V}^T \mathbf{Y}) \text{ s.t. } \max_{\substack{\mathbf{A}\mathbf{u} \succcurlyeq \mathbf{0} \\ \mathbf{u} \in \mathcal{B}_2}} \|\mathbf{V}^T \mathbf{A}\mathbf{u}\|_2 \leq 1, \ \forall \mathbf{u} \in \mathcal{B}_2. \tag{29}$$

Then, the the corresponding bidual problem is

$$\min_{\{\mathbf{w}_{1j}, \mathbf{w}_{2j}\}_{j=1}^m} \sum_{j=1}^m \|\mathbf{w}_{1j}\|_2 \|\mathbf{w}_{2j}\|_2 \text{ s.t. } \sum_{j=1}^m \mathbf{A}\mathbf{w}_{1j}\mathbf{w}_{2j}^T = \mathbf{Y}, \ \mathbf{A}\mathbf{w}_{1j} \succcurlyeq \mathbf{0}, \ \forall j, \tag{30}$$

which can be stated as a convex optimization problem as

$$\min_{\mathbf{M} \in \mathcal{C}} \|\mathbf{M}\|_* \text{ s.t. } \mathbf{A}\mathbf{M} = \mathbf{Y}, \tag{31}$$

as long as $m \geq m^* := \operatorname{rank}(\mathbf{M}^*)$ where $\mathbf{M}^*$ is an optimal solution, $\mathcal{C} := \operatorname{conv}\{\mathbf{w}_1 \mathbf{w}_2^T : \mathbf{A}\mathbf{w}_1 \succcurlyeq \mathbf{0}, \mathbf{w}_1 \in \mathbb{R}^d, \mathbf{w}_2 \in \mathbb{R}^o\}$, $\| \cdot \|_*$ denotes the nuclear norm, and conv represents the convex hull of a set. We remark that the problem in (31) resembles convex semi non-negative matrix factorizations, such as the ones studied in Ding et al. (2008); Sahiner et al. (2021b). These problems are not tractable in polynomial time in the worst case. For instance taking $\mathbf{A} = \mathbf{I}_n$ simplifies to a copositive program, which is NP-hard for arbitrary $\mathbf{Y}$.

4.5.5 ℓ_1 regularized version of vector output case

Notice that since (25) 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 (23) so that extreme points can be efficiently achieved using convex optimization tools. Consider the following optimization problem

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

Then using the scaling trick in Lemma 1, we obtain the following

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

which has the following dual form

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

and an optimal $\mathbf{U}$ satisfies

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

where $\mathbf{V}^*$ is the optimal dual variable. Note that 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 $k \in [o]$

$$\hat{\mathbf{u}}_{k,1} = \operatorname*{argmax}_{\mathbf{u}} \mathbf{v}_k^T\mathbf{Au} \text{ s.t. } \mathbf{Au} \succcurlyeq \mathbf{0}, \|\mathbf{u}\|_2 \leq 1$$
$$\hat{\mathbf{u}}_{k,2} = \operatorname*{argmin}_{\mathbf{u}} \mathbf{v}_k^T\mathbf{Au} \text{ s.t. } \mathbf{Au} \succcurlyeq \mathbf{0}, \|\mathbf{u}\|_2 \leq 1,$$

where $\mathbf{v}_k$ is the k^{th} 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

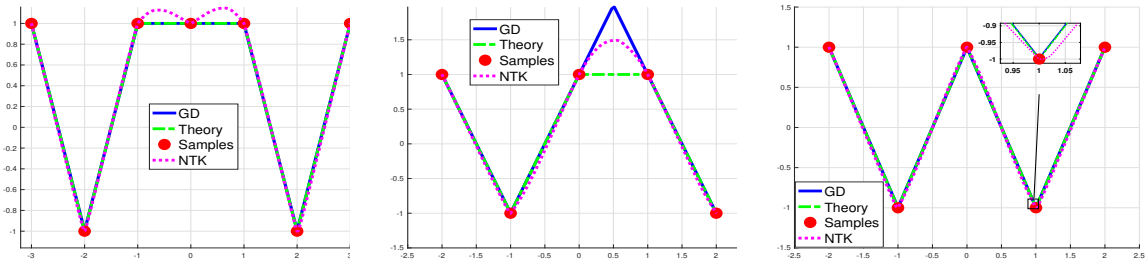
$$\min_{\theta\in\Theta} \|\mathbf{W}\|_0 \text{ s.t. } f_\theta(\mathbf{A}) = \mathbf{Y}, \|\mathbf{u}_j\|_2 = 1, \forall j.$$

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

Lemma 12 *Suppose that $n \leq d$, $\mathbf{A}$ 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* (13) *is given by*

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

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



(a) (b) (c)

Figure 9: One dimensional regression task with square loss, where we apply GD, NTK, and our approach in Corollary 8.

Lemma 13 *We have ℓ_1-ℓ_0 equivalence if the following condition holds*

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

5. Comparison with Neural Tangent Kernel

Here, we first briefly discuss the recently introduced NTK (Jacot et al., 2018) and other connections to kernel methods. We then compare this approach with our exact characterization in terms of a kernel matrix in Corollary 8.

The connection between kernel methods and infinitely wide NNs has been extensively studied (Neal, 1996; Matthews et al., 2018; Lee et al., 2017). 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 a 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 simultaneously. Therefore, Jacot et al. (2018) introduced a new kernel method, 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 interpreted 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 Chizat and Bach (2018) for details and limitations of the NTK framework. For one dimensional problems, this kernel characterization can be written as follows (Bietti and Mairal, 2019)[9]

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

9. We provide the NTK formulation without a bias term to simplify the presentation. The expression for a case with bias can be found in Williams et al. (2019).

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

$$\min \|\mathbf{w}\|_2^2 \text{ s.t. } \mathbf{K}\mathbf{w} = \mathbf{y}. \tag{32}$$

We first note that our approach in (17) is different than the NTK approach in (32) in terms of kernel construction and objective function.

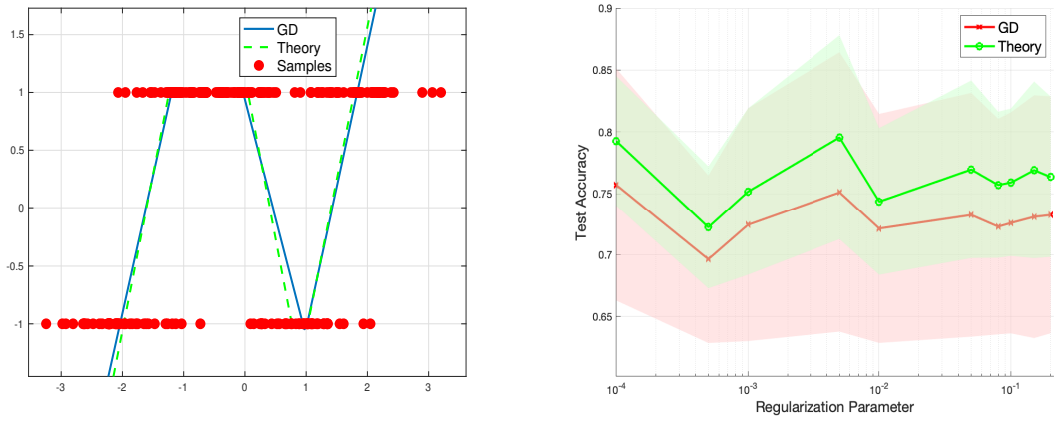
In order to compare the performance of (17), (32) and GD, we perform experiments on one dimensional datasets. In Figure 9, we observe that NTK outputs smoother functions compared to GD and our approach. Particularly, in Figure 9a and 9b, we clearly see that the output of NTK is not a piecewise linear function unlike our approach and GD. Moreover, even though the output of NTK looks like a piecewise linear function in Figure 9c, 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 note that optimal solution might not be unique as proven in Proposition 1 and Figure 7, which explains why GD converges to a solution with a kink in the middle of two data points while the kinks obtained by our approach exactly aligns with the data points.

6. Numerical Experiments

We first consider a binary classification experiment using hinge loss on a synthetic dataset[10]. 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 11, i.e., denoted as Theory and GD on these datasets. In Figure 10, 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 10a, 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 (LeCun) and CIFAR-10 (Krizhevsky et al., 2014). In order to verify our results in Theorem 15, we run 5 SGD trials with independent initializations for the network parameters, where we use subsampled versions of the datasets. As illustrated in Figure 11 and 12, the network constructed using the closed-form solution achieves the lowest training objective and highest test accuracy for both datasets. In addition to our closed-form solutions, we also propose a convex cutting plane based approach to optimize ReLU networks. In Table

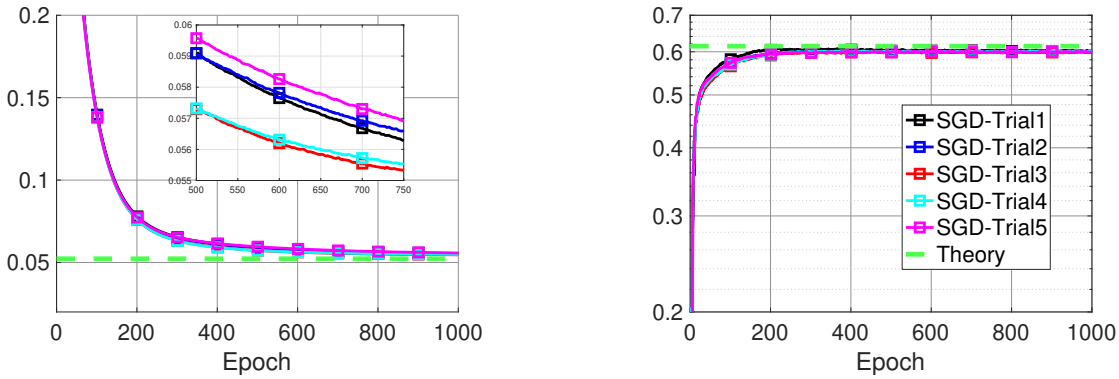
10. We provide further details on the numerical experiments in Appendix 8.



(a) Visualization of the samples and the function fit.

(b) Test accuracy with 90% confidence band.

Figure 10: Binary classification using hinge loss, where we apply GD and our approach in Theorem 11, i.e., denoted as Theory.

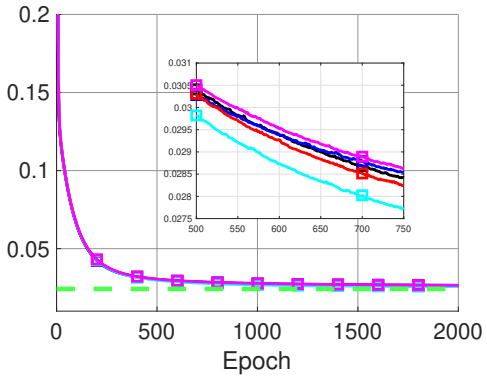


(a) MNIST-Training objective

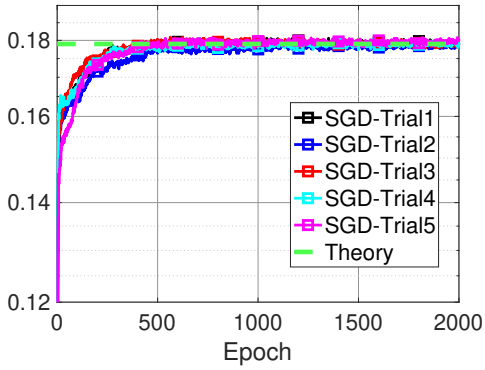
(b) MNIST-Test accuracy

Figure 11: Training and test performance of 5 independent SGD trials on whitened and sampled MNIST, where $(n, d) = (200, 250)$, $K = 10$, $\beta = 10^{-3}$, $m = 100$ and we use squared loss with one hot encoding. For the method denoted as Theory, we use the layer weights in Theorem 15.

1, we observe that our approach denoted as `Convex`, which is completely based on convex optimization, outperforms the non-convex backpropagation based approach. Note that we use the full datasets for this experiment. Furthermore, we use an alternative approach, denoted as `Convex-RF` in Table 1 which uses (10) on image patches to obtain 40k filters, e.g., Figure 13 (further details are provided in the next section). 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 datasets, namely Bank, Boston Housing, California Housing, Eleva-



(a) CIFAR10-Training objective



(b) CIFAR10-Test accuracy

Figure 12: Training and test performance of 5 independent SGD trials on whitened and sampled CIFAR-10, where $(n, d) = (60, 60)$, $K = 10$, $\beta = 10^{-3}$, $m = 100$ and we use squared loss with one hot encoding. For Theory, we use the layer weights in Theorem 15.

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%**	-	-	-	-	-	-

tors, Stock (Torgo), and the Twenty Newsgroups text classification dataset (Mitchell and Learning, 1997). In Table 1, we provide the test errors for each approach. Here, our convex approach outperforms the backpropagation, and the one layer NN in each case.

6.1 Unsupervised and interpretable training using extreme points: `Convex-RF`

For this approach, we use the convolutional neural net architecture in Coates and Ng (2012). However, instead of using random filters as in Zhang et al. (2016) or applying the k-means algorithm as in Coates and Ng (2012), 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 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 Coates and Ng (2012). 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.

Figure 13: Extreme points found by (10) applied on image patches of CIFAR-10 yield filters used in the `Convex-RF` algorithm. Note that the extreme points visually correspond to predictive image patches.

Algorithm 2 `Convex-RF`

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

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

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

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

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

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

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

Remark 5 *Note that the ZCA whitening step plays a crucial role for similar implementations in Coates and Ng (2012); Shankar et al. (2020); Thiry et al. (2021). We believe that this is mainly due to the fact that these approaches rely on either direct distance computations or inner products with the samples, where whitening can enable more informative features, i.e., either the distance computation or inner product, by transforming the sam-*

ples (see Appendix A of Thiry et al. (2021)). However, in our case, the hidden neurons constructed from the data samples are inherently informative and normalized as detailed in Step 9 of Algorithm 2. Particularly, since we compute the distance of each sample to the convex hull of the remaining samples instead of computing pairwise distances, our approach is much more robust against the correlations among patches. In addition, due to the scaling in Lemma 1, the hidden neurons are normalized such that inner products tend to be drastically less dependent on the correlations. Therefore, whitening don't change our results and this is why we remark that the whitening step is optional in Algorithm 2.

7. Concluding Remarks

We studied two-layer ReLU networks and introduced a convex analytic framework based on duality to characterize a set of optimal solutions to the regularized training problem. Our analysis showed that optimal solutions can be exactly characterized as the extreme points of a convex set. More importantly, these extreme points yield simple structures at the network output, which explains why ReLU networks fit structured functions, e.g., a linear spline interpolation for one dimensional datasets. Moreover, by establishing a relation with minimum cardinality problems in compressed sensing, we even provided closed-form solutions for the optimal hidden layer weights in various practically relevant scenarios such as problems involving rank-one or spike-free data matrices. Therefore, for such cases, one can directly use these closed-form solutions and then train only the output layer, i.e., a linear layer, in a similar fashion to kernel regression/classification problems. However, unlike the existing kernel methods, e.g., NTK (Jacot et al., 2018), our approach is regularized by ℓ_1-norm encouraging sparse solutions. Thus, we are able to match the performance of a classical finite width ReLU network trained with SGD, for which existing kernel methods fail to provide a satisfactory approximation, by solving a linear minimum ℓ_1-norm problem with a fixed matrix. Additionally, we provided an iterative algorithm based on the cutting plane method to optimize the network parameters for problems with arbitrary data and then proved its convergence to the global optimum under certain assumptions.

In the light of our results, there are multiple future research directions, which we want to mention as open research problems. First of all, our analysis reveals that the original non-convex training problem has a geometrical structure that can be fully characterized by convex duality. Under certain technical conditions such as spike-freeness, whitened, or rank-one matrices, this geometry is easily understood by closed-form expressions of the extreme points. We conjecture that one can also utilize convex duality to understand the optimization landscape and extraordinary generalization abilities of deep networks. For instance, the recent findings in Lacotte and Pilanci (2020); Lederer (2020) regarding the global optimality of all local minima in sufficiently wide networks can be understood through the lens of convex analysis. In addition to this, our approach explains why NTK (Jacot et al., 2018) and other kernel methods fail to recover the exact training dynamics of finite width ReLU networks. We also show that one can obtain closed-form solutions for all network parameters, i.e, hidden and output layer weights, in some special cases such as problems with rank-one or whitened data matrices. Hence, there is no need to train a fully connected two-layer ReLU network via SGD in these cases. Based on these observations, we believe that the active learning regime as coined in Chizat and Bach (2018), or a new

transition regime, where hidden neurons actively learn useful features that generalize well can be analytically characterized. Finally, one can also extend our polynomial-time convex formulations for spike-free problems to develop efficient solvers to globally optimize deeper networks via layerwise learning. Even though certain parts of our analysis are restricted certain classes of data matrices such as spike-free, whitened, rank-one or one dimensional, we conjecture that a similar convex analytic framework can be developed for arbitrary data distributions, alternative network architectures and deeper networks. After a preliminary version of this manuscript appeared in Ergen and Pilanci (2020a), our follow-up work (Ergen and Pilanci, 2020b; Pilanci and Ergen, 2020; Ergen and Pilanci, 2021; Sahiner et al., 2021a) aimed to address some of these questions via the convex analytic tools developed in this present work.

Acknowledgments

This work was partially supported by the National Science Foundation under grants IIS-1838179, ECCS-2037304 and the Army Research Office.

Appendix

Table of Contents

In this section, we present proofs of the main results and further details on the algorithms and numerical results.

8. 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 datasets. For comparison, we also include the performance of a two-layer NN trained with the backpropagation 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 (Krogh and Hertz, 1992). In addition to this, we use the cutting plane based algorithm along with the neurons in (10) for our convex approach. In order to solve the convex optimization problems in our approach, we use CVX (Grant and Boyd, 2014). However, notice that when dealing with large datasets, e.g., CIFAR-10, plain CVX solvers might need significant amount of memory. In order to circumvent these issues, we use SPGL1 (van den Berg and Friedlander, 2007) and SuperSCS (Themelis and Patrinos, 2019) for large datasets. We also remark that all the datasets we use are publicly available and further information, e.g., training and test sizes, can be obtained through the provided references (LeCun; Krizhevsky et al., 2014; Torgo; new). 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.

9. 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 1

$$\max_{\mathbf{v}:\mathbf{1}^T\mathbf{v}=0} \mathbf{v}^T\mathbf{y} \text{ s.t. } \left|\mathbf{v}^T\left(\mathbf{A}\mathbf{u}+b\mathbf{1}\right)_+\right| \leq 1\,, \forall \mathbf{u}\in\mathcal{B}_2, \forall b\in\mathbb{R} \tag{33}$$

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 Goberna and López-Cerdá (1998).

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

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

10. Infinite size neural networks

Here we briefly review infinite size, i.e., infinite width, two-layer NNs (Bach, 2017). We refer the reader to Bengio et al. (2006); Wei et al. (2018) 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 (Rudin, 1964). 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 $\phi_{\mathbf{u}}(\mathbf{x}) = \left(\mathbf{x}^T\mathbf{u}\right)_+$. 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 (4) corresponds to

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

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

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

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

11. Proofs of the main results

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

11.1 Proofs for the results in Section 2

Proof of Lemma 1 We first note that similar proofs are also presented in Neyshabur et al. (2014); Savarese et al. (2019); Ergen and Pilanci (2019b, 2020a,b,c); Pilanci and Ergen (2020); Ergen et al. (2021); Gupta et al. (2021). 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, (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 $\|\mathbf{w}\|_1$. ■

Proof of Lemma 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. ■

Proof of Lemma 3 The first condition immediately implies that $\{\left(\mathbf{A}\mathbf{u}\right)_+ | \mathbf{u} \in \mathcal{B}_2\} \subseteq \mathbf{A}\mathcal{B}_2$. Since we also have $\{\left(\mathbf{A}\mathbf{u}\right)_+ | \mathbf{u} \in \mathcal{B}_2\} \subseteq \mathbb{R}^n_+$, it holds that $\{\left(\mathbf{A}\mathbf{u}\right)_+ | \mathbf{u} \in \mathcal{B}_2\} \subseteq \mathbf{A}\mathcal{B}_2 \cap \mathbb{R}^n_+$. 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

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

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 $\mathbf{I}_n - \mathbf{A}\mathbf{A}^\dagger = \mathbf{0}$. ■

Proof of Lemma 4 We have

$$\begin{aligned}
\max_{\mathbf{u}\,:\,\|\mathbf{u}\|_2\leq 1} \|\mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\big(\mathbf{A}\mathbf{u}\big)_+\|_2 &\leq \sigma_{\max}(\mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}) \max_{\mathbf{u}\,:\,\|\mathbf{u}\|_2\leq 1} \|\big(\mathbf{A}\mathbf{u}\big)_+\|_2 \\
&= \sigma^{-1}_{\min}(\mathbf{A}) \max_{\mathbf{u}\,:\,\|\mathbf{u}\|_2\leq 1} \|\big(\mathbf{A}\mathbf{u}\big)_+\|_2 \\
&\leq \sigma^{-1}_{\min}(\mathbf{A}) \max_{\mathbf{u}\,:\,\|\mathbf{u}\|_2\leq 1} \|\mathbf{A}\mathbf{u}\|_2 \\
&\leq \sigma^{-1}_{\min}(\mathbf{A})\sigma_{\max}(\mathbf{A}) \\
&\leq 1\,.
\end{aligned}$$

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

Proof of Lemma 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, $\big(\mathbf{A}\mathbf{u}\big)_+ = \mathbf{c}\big(\mathbf{a}^T\mathbf{u}\big)_+$. If $\big(\mathbf{a}^T\mathbf{u}\big)_+ = 0$, then we can select $\mathbf{z} = \mathbf{0}$ to satisfy the spike-free condition $\big(\mathbf{c}\mathbf{a}^T\mathbf{u}\big)_+ = \mathbf{A}\mathbf{z}$. If $\big(\mathbf{a}^T\mathbf{u}\big)_+ \neq 0$, then $\big(\mathbf{A}\mathbf{u}\big)_+ = \mathbf{c}\mathbf{a}^T\mathbf{u} = \mathbf{A}\mathbf{u}$, where the spike-free condition can be trivially satisfied with the choice of $\mathbf{z} = \mathbf{u}$. ■

Proof of Lemma 6 The extreme point along the direction of $\mathbf{v}$ can be found as follows

$$\operatorname*{argmax}_{u,b} \sum_{i=1}^n v_i(a_i u + b)_+ \text{ s.t. } |u| = 1, \tag{34}$$

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 (34) as

$$\operatorname*{argmax}_{u,b} \sum_{i\in\mathcal{S}} v_i(a_i u + b) \text{ s.t. } a_i u + b \geq 0, \forall i \in \mathcal{S}, a_j u + b \leq 0, \forall j \in \mathcal{S}^c, |u| = 1. \tag{35}$$

In (35), 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, (35) reduces to finding the optimal bias. We note that due to the constraints in (35), $-a_i \leq b \leq -a_j, \forall i \in \mathcal{S}, \forall j \in \mathcal{S}^c$. 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

$$b_v = \begin{cases} \max_{i\in\mathcal{S}}(-a_i), & \text{if } \sum_{i\in\mathcal{S}} v_i \leq 0 \\ \min_{j\in\mathcal{S}^c}(-a_j), & \text{otherwise} \end{cases}. \tag{36}$$

Similar arguments also hold for $u = -1$ and the argmin version of (34). Note that when $\sum_{i\in\mathcal{S}} v_i = 0$, the value of the bias does not change the objective in (35). 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. ■

Proof of Lemma 7 For the extreme point in the span of $\mathbf{e}_i$, we need to solve the following optimization problem

$$\underset{\mathbf{u},b}{\text{argmax}}(\mathbf{a}_i^T\mathbf{u}+b) \text{ s.t. } \mathbf{a}_j^T\mathbf{u}+b \leq 0, \forall i \neq j, \|\mathbf{u}\|_2 = 1. \tag{37}$$

Then the Lagrangian of (37) is

$$L(\boldsymbol{\lambda}, \mathbf{u}, b) = \mathbf{a}_i^T\mathbf{u} + b - \sum_{\substack{j=1 \\ j\neq i}}^{n} \lambda_j(\mathbf{a}_j^T\mathbf{u}+b), \tag{38}$$

where we do not include the unit norm constraint for $\mathbf{u}$. For (38), $\boldsymbol{\lambda}$ must satisfy $\boldsymbol{\lambda} \succcurlyeq \mathbf{0}$ and $\mathbf{1}^T\boldsymbol{\lambda} = 1$. With these specifications, the problem can be written as

$$\min_{\boldsymbol{\lambda}} \max_{\mathbf{u}} \mathbf{u}^T \left(\mathbf{a}_i - \sum_{\substack{j=1 \\ j\neq i}}^{n} \lambda_j \mathbf{a}_j \right) \text{ s.t. } \boldsymbol{\lambda} \succcurlyeq \mathbf{0}, \mathbf{1}^T\boldsymbol{\lambda} = 1, \|\mathbf{u}\|_2 = 1. \tag{39}$$

Since the $\mathbf{u}$ vector that maximizes (39) is the normalized version of the term inside the parenthesis above, the problem reduces to

$$\min_{\boldsymbol{\lambda}} \left\| \mathbf{a}_i - \sum_{\substack{j=1 \\ j\neq i}}^{n} \lambda_j \mathbf{a}_j \right\|_2 \text{ s.t. } \boldsymbol{\lambda} \succcurlyeq \mathbf{0}, \mathbf{1}^T\boldsymbol{\lambda} = 1. \tag{40}$$

After solving the convex problem (40) 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 (37). ■

Proof of Lemma 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

$$\underset{\mathbf{u},b}{\text{argmax}}\, \boldsymbol{\alpha}^T(\mathbf{A}\mathbf{u} + b\mathbf{1})_+ \text{ s.t. } \|\mathbf{u}\|_2 \leq 1 \tag{41}$$

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 (41) 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

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

■

11.2 Proofs for the results in Section 3

Proof of Theorem 1 and Corollary 2

We first note that the dual of (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

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

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

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

Since the set $\|(\mathbf{A}\mathbf{U})_+^T \mathbf{v}\|_\infty \leq 1$ is closed, the function $\Phi(\mathbf{v}, \mathbf{U}) = \mathbf{v}^T \mathbf{y} + \mathcal{I}(\|(\mathbf{A}\mathbf{U})_+^T \mathbf{v}\|_\infty \leq 1)$ 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 (12), which establishes a lower bound D^* for the above problem:

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

We now show that strong duality holds for infinite size NNs. The dual of the semi-infinite program in (12) is given by (see Section 2.2 of Goberna and López-Cerdá (1998) and also Bach (2017))

$$\begin{aligned} &\min \|\boldsymbol{\mu}\|_{TV} \\ &\text{s.t.} \int_{\mathbf{u}\in\mathcal{B}_2} (\mathbf{A}\mathbf{u})_+ d\boldsymbol{\mu}(\mathbf{u}) = \mathbf{y}, \end{aligned}$$

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 10, and therefore strong duality holds. We also remark that even though the above problem involves an infinite dimensional integral form, by Caratheodory's theorem, this integral form can be represented as a finite summation with at most $n+1$ Dirac delta functions (Rosset et al., 2007). Next we invoke the semi-infinite optimality conditions for the dual problem in (12), in particular we apply Theorem 7.2 of Goberna and López-Cerdá (1998). We first define the set

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

Note that $\mathbf{K}$ is the union of finitely many convex closed sets, since the function $\big(\mathbf{Au}\big)_+$ can be expressed as the union of finitely many convex closed sets. Therefore the set $\mathbf{K}$ is closed. By Theorem 5.3 of Goberna and López-Cerdá (1998), this implies that the set of constraints in (12) forms a Farkas-Minkowski system. By Theorem 8.4 of Goberna and López-Cerdá (1998), 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 (12). The optimality conditions in Theorem 7.2 of Goberna and López-Cerdá (1998) implies that $\mathbf{y} = \big(\mathbf{AU}^*\big)_+ \mathbf{w}^*$ for some vector $\mathbf{w}^*$. Since the primal and dual values are equal, we have $\mathbf{v}^{*T}\mathbf{y} = \mathbf{v}^{*T}\big(\mathbf{AU}^*\big)_+ \mathbf{w}^* = \|\mathbf{w}^*\|_1$, which shows that the primal-dual pair $(\{\mathbf{w}^*, \mathbf{U}^*\}, \mathbf{v}^*)$ is optimal. Thus, the optimal neuron weights $\mathbf{U}^*$ satisfy $\|(\mathbf{AU}^*)_+^T \mathbf{v}^*\|_\infty = 1$. ■

Proof of Theorem 2 We first assume that zero training error can be achieved with m_1 neurons. Then, we obtain the dual of (4) with $m = m_1$

$$P_f^* = \min_{\theta \in \Theta \setminus \{\mathbf{w}, m\}} \max_{\mathbf{v}} \mathbf{v}^T \mathbf{y} \text{ s.t. } \|(\mathbf{AU})_+^T \mathbf{v}\|_\infty \leq 1,\ \|\mathbf{u}_j\|_2 \leq 1, \forall j \in [m_1]. \tag{42}$$

Exchanging the order of min and max establishes a lower bound for (42)

$$P_f^* \geq D_f^* = \max_{\mathbf{v}} \min_{\theta \in \Theta \setminus \{\mathbf{w}, m\}} \mathbf{v}^T \mathbf{y} + \mathcal{I}(\|(\mathbf{AU})_+^T \mathbf{v}\|_\infty \leq 1),\ \text{s.t. } \|\mathbf{u}_j\|_2 \leq 1, \forall j \in [m_1]. \tag{43}$$

If we denote the optimal parameters to (42) 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 (42), 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

$$\begin{aligned} P_f^* = \min_{\theta \in \Theta \setminus \{\mathbf{w}, m\}} \max_{\mathbf{v}} \mathbf{v}^T \mathbf{y} \quad & \geq \quad \min_{\theta \in \Theta \setminus \{\mathbf{w}\}} \max_{\mathbf{v}} \mathbf{v}^T \mathbf{y} \\ \text{s.t. } \|(\mathbf{A}\mathbf{U})_+^T \mathbf{v}\|_\infty \leq 1,\ \|\mathbf{u}_j\|_2 \leq 1, \forall j \in [m_1] \quad & \text{s.t. } \|(\mathbf{A}\mathbf{U})_+^T \mathbf{v}\|_\infty \leq 1,\ \|\mathbf{u}_j\|_2 \leq 1, \forall j \\ & \geq \quad \max_{\mathbf{v}} \min_{\theta \in \Theta \setminus \{\mathbf{w}\}} \mathbf{v}^T \mathbf{y} \\ & \quad \text{s.t. } \|(\mathbf{A}\mathbf{U})_+^T \mathbf{v}\|_\infty \leq 1,\ \|\mathbf{u}_j\|_2 \leq 1, \forall j \\ & = \quad \max_{\mathbf{v}} \min_{\theta \in \Theta \setminus \{\mathbf{w}, m\}} \mathbf{v}^T \mathbf{y} \\ & \quad \text{s.t. } (\|(\mathbf{A}\mathbf{U})_+^T \mathbf{v}\|_\infty \leq 1,\ \|\mathbf{u}_j\|_2 \leq 1, \forall j \in [m_1] \\ & = \quad D_f^* = D^* \end{aligned} \tag{44}$$

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 (43). 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 (42) do not change the objective value. Therefore, by (44), we prove that weak duality holds for a finite width NN, i.e., $P_f^* \geq P^* \geq D_f^* = D^*$.

■

Proof of Theorem 3 By Caratheodory's theorem (see Rosset et al. (2007)), the number of constraints active in the dual problem is bounded by $n+1$. Suppose this number is m^*, where $m^* \leq n+1$. Thus, we can construct a weight matrix $\mathbf{U}_e \in \mathbb{R}^{d \times m^*}$ that consists of all the extreme points. Next, the dual of (4) with $\mathbf{U} = \mathbf{U}_e$

$$D_f^* = \max_{\mathbf{v}} \mathbf{v}^T \mathbf{y} \text{ s.t. } \|(\mathbf{A}\mathbf{U}_e)_+^T \mathbf{v}\|_\infty \leq 1, \tag{45}$$

Consequently, we have

$$\begin{aligned} P^* = \min_{\theta \in \Theta \setminus \{\mathbf{w}\}} \max_{\mathbf{v}} \mathbf{v}^T \mathbf{y} \quad & \geq \quad \max_{\mathbf{v}} \min_{\theta \in \Theta \setminus \{\mathbf{w}\}} \mathbf{v}^T \mathbf{y} \\ \text{s.t } \|(\mathbf{A}\mathbf{U})_+^T \mathbf{v}\|_\infty \leq 1,\ \|\mathbf{u}_j\|_2 \leq 1, \forall j \quad & \quad \text{s.t. } \|(\mathbf{A}\mathbf{U})_+^T \mathbf{v}\|_\infty \leq 1,\ \|\mathbf{u}_j\|_2 \leq 1, \forall j \\ & = \quad \max_{\mathbf{v}} \mathbf{v}^T \mathbf{y} \\ & \quad \text{s.t. } (\|(\mathbf{A}\mathbf{U}_e)_+^T \mathbf{v}\|_\infty \leq 1 \\ & = \quad D_f^* = D^* \end{aligned} \tag{46}$$

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 2 and our observation above, i.e., neurons that are not the extreme point of the inequality in (45) 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

$$\begin{aligned} P_f^* = \min_{\theta \in \Theta \setminus \{\mathbf{U}, m\}} \|\mathbf{w}\|_1 \quad & \geq \quad P^* = \min_{\theta \in \Theta} \|\mathbf{w}\|_1 \\ \text{s.t. } (\mathbf{A}\mathbf{U}_e)_+ \mathbf{w} = \mathbf{y} \quad & \quad \text{s.t. } (\mathbf{A}\mathbf{U})_+ \mathbf{w} = \mathbf{y},\ \|\mathbf{u}_j\|_2 \leq 1, \forall j, \end{aligned} \tag{47}$$

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 (47) 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 (46), we prove that strong duality holds for a finite width NN, i.e., $P_f^* = P^* = D^* = D_f^*$. ■

Proof of Proposition 1

We first note that the conditions related to the range of bias values that lead to non-uniqueness directly follows from Proof of Lemma 6. Hence here, we particularly examine the problem in (4) when we have a one dimensional dataset, i.e., $\{a_i, y_i\}_{i=1}^n$, to provide analytic forms for the counter-examples depicted in Figure 7. Then, (4) can be modified as

$$\min_{\theta \in \Theta} \|\mathbf{w}\|_1 \text{ s.t. } (\mathbf{a}\mathbf{u}^T + \mathbf{1}\mathbf{b}^T)_+ \mathbf{w} = \mathbf{y}, |u_j| \leq 1, \forall j. \tag{48}$$

Then, using Lemma 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, we can rewrite (48) as

$$\min_{\mathbf{w}} \|\mathbf{w}\|_1 \text{ s.t. } \mathbf{A}_e \mathbf{w} = \mathbf{y}. \tag{49}$$

In the following, we first derive optimality conditions for (49) and then provide an analytic counter example to disprove uniqueness. Then, we also follow the same steps for the regularized version of (49).

Equality constraint: The optimality conditions for (49) are

$$\begin{aligned} &\mathbf{A}_e \mathbf{w}^* = \mathbf{y} \\ &\mathbf{A}_{e,s}^T \mathbf{v}^* + \text{sign}(\mathbf{w}_s^*) = 0 \\ &\|\mathbf{A}_{e,s^c}^T \mathbf{v}^*\|_\infty \leq 1, \end{aligned} \tag{50}$$

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 (50).

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 (49) 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 (49) 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 (50).

Regularized case: The regularized version of (49) is as follows

$$\min_{\mathbf{w}} \beta\|\mathbf{w}\|_1 + \frac{1}{2n}\|\mathbf{A}_e\mathbf{w} - \mathbf{y}\|_2^2, \tag{51}$$

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

$$\begin{aligned} &\frac{1}{n}\mathbf{A}_{e,s}^T(\mathbf{A}_e\mathbf{w}^* - \mathbf{y}) + \beta\text{sign}(\mathbf{w}_s^*) = 0 \\ &\|\mathbf{A}_{e,s^c}^T(\mathbf{A}_e\mathbf{w}^* - \mathbf{y})\|_\infty \leq \beta n, \end{aligned} \tag{52}$$

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

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

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 (51) 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 (52). We also provide a visualization for the output functions of each solution in Figure 7, namely Solution1 and Solution2.

Remark 1 *In fact, there exist infinitely many solutions to the regularized training problem discussed 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 $0 \leq c \leq 1$. 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 7. These solutions also achieve the same objective value and their last layer weights are as

follows

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

This numerical observation can also be explained via our extreme point characterization in (36)*, where the optimal bias can take one of infinitely many possible values in a certain interval when* $\sum_{i\in\mathcal{S}} v_i = 0$*.*

■

Proof of Corollary 5 Given $\mathbf{A} = \mathbf{c}\mathbf{a}^T$, all possible extreme points can be characterized as follows

$$\begin{aligned} \underset{b,\mathbf{u}:\|\mathbf{u}\|_2=1}{\operatorname{argmax}} |\mathbf{v}^T\big(\mathbf{A}\mathbf{u} + b\mathbf{1}\big)_+| &= \underset{b,\mathbf{u}:\|\mathbf{u}\|_2=1}{\operatorname{argmax}} |\mathbf{v}^T\big(\mathbf{c}\mathbf{a}^T\mathbf{u} + b\mathbf{1}\big)_+| \\ &= \underset{b,\mathbf{u}:\|\mathbf{u}\|_2=1}{\operatorname{argmax}} \Big| \sum_{i=1}^n v_i\big(c_i\mathbf{a}^T\mathbf{u} + b\big)_+ \Big| \end{aligned}$$

which can be equivalently stated as

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

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

Proof of Lemma 9 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

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

Then note that

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

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

Proof of Lemma 10 We first provide the optimality conditions for the convex program in the following proposition:

Proposition 3 *Let* $\mathbf{U}$ *be a weight matrix for* (4)*. Then,* $\mathbf{U} \in \mathbb{R}^{d\times m}$ *is an optimal solution for the regularized training problem if*

$$\exists \boldsymbol{\alpha} \in \mathbb{R}^n, \mathbf{w} \in \mathbb{R}^m \ s.t. \ \big(\mathbf{A}\mathbf{U}\big)_+\mathbf{w} = \mathbf{y}, \quad \big(\mathbf{A}\mathbf{U}\big)_+^T\boldsymbol{\alpha} = sign(\mathbf{w}) \tag{53}$$

and

$$\max_{\mathbf{u}\,:\,\|\mathbf{u}\|_2\leq 1} |\boldsymbol{\alpha}^T(\mathbf{A}\mathbf{u})_+| \leq 1\,. \tag{54}$$

These conditions follow from linear semi-infinite optimality conditions given in Theorem 7.1 and 7.6 of Goberna and López-Cerdá (1998) for Farkas-Minkowski systems. Then the proof Lemma 10 directly follows from the solution of minimum cardinality problem given in Lemma 9.

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

$$\max_{|\mathbf{v}^T(\mathbf{A}\mathbf{u})_+|\leq 1,\ \forall \mathbf{u}\in\mathcal{B}_2} \mathbf{v}^T\mathbf{y} \tag{55}$$

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

$$\max_{|\mathbf{v}^T(\mathbf{u}')_+|\leq 1,\ \forall \mathbf{u}'\in\mathcal{B}_2} \mathbf{v}^T\mathbf{y} = \max_{\|(\mathbf{v})_+\|_2\leq 1,\ \|(-\mathbf{v})_+\|_2\leq 1} \mathbf{v}^T\mathbf{y}\,. \tag{56}$$

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 (56). Now note that

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

Therefore the right-hand-side of (56) 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

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

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. ■

Proof of Proposition 2 Since the constraint in (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 Goberna and López-Cerdá (1998). ■

Proof of Theorem 5 Given a vector $\mathbf{u}$ we partition $\mathbf{A}$ according to the subset $S = \{i|\mathbf{a}_i^T\mathbf{u} \geq 0\}$, 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 $\big(\mathbf{A}\mathbf{u}\big)_+$ as follows

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

Then we use the block matrix pseudo-inversion formula (Baksalary and Baksalary, 2007)

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

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

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

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

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

■

Proof of Theorem 6 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 (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. ■

Proof of Theorem 7 Let us define the distance of the i^{th} sample vector to the convex hull of the remaining sample vectors as d_i

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

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

Using Gordon's escape from a mesh theorem (Gordon, 1988; Ledoux and Talagrand, 2013), we obtain the following lower-bound on the expectation of d_i

$$\begin{aligned}
\mathbb{E}\, d_i &\geq \mathbb{E} \min_{\substack{\mathbf{z}\in\mathbb{R}^n: \\ \sum_{j\neq i} z_j=1 \\ z_j\geq 0, z_i=-1}} \max_{\mathbf{v}:\|\mathbf{v}\|_2\leq 1} \mathbf{h}^T\mathbf{v}\|\mathbf{z}\|_2 + \mathbf{z}^T\mathbf{g} \\
&= \mathbb{E} \min_{\substack{\mathbf{z}\in\mathbb{R}^n: \\ \sum_{j\neq i} z_j=1 \\ z_j\geq 0, z_i=-1}} ||\mathbf{h}||_2\|\mathbf{z}\|_2 + \mathbf{z}^T\mathbf{g} \\
&\geq \sqrt{d}\|\mathbf{z}\|_2 - \mathbb{E} \max_{j\in[n], j\neq i} g_j + g_i \\
&\geq \frac{\sqrt{d}}{\sqrt{n}} - \sqrt{2\log(n-1)}\,, \qquad (57)
\end{aligned}$$

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 (Ledoux and Talagrand, 2013). Therefore, the expected distance of the i^{th} sample to the convex hull is guaranteed to be positive whenever $d > 2n\log(n-1)$. Note that the lower bound (57) 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

$$\begin{aligned}
\min_{\substack{\mathbf{z}\in\mathbb{R}^n: \\ \sum_{j\neq i} z_j=1 \\ z_j\geq 0, z_i=-1}} \|\mathbf{A}^T\mathbf{z}\|_2 - \min_{\substack{\mathbf{z}\in\mathbb{R}^n: \\ \sum_{j\neq i} z_j=1 \\ z_j\geq 0, z_i=-1}} \|\tilde{\mathbf{A}}^T\mathbf{z}\|_2 &\leq \min_{\substack{\mathbf{z}\in\mathbb{R}^n: \\ \sum_{j\neq i} z_j=1 \\ z_j\geq 0, z_i=-1}} \|\big(\mathbf{A}-\tilde{\mathbf{A}}^T\big)\mathbf{z}\|_2 \\
&\leq \|(\mathbf{A}-\tilde{\mathbf{A}})\|_2 \max_{\substack{\mathbf{z}\in\mathbb{R}^n: \\ \sum_{j\neq i} z_j=1 \\ z_j\geq 0, z_i=-1}} \|\mathbf{z}\|_2 \\
&\leq \|(\mathbf{A}-\tilde{\mathbf{A}})\|_F \max_{\substack{\mathbf{z}\in\mathbb{R}^n: \\ \sum_{j\neq i} z_j=1 \\ z_j\geq 0, z_i=-1}} \|\mathbf{z}\|_1 \\
&\leq 2\|(\mathbf{A}-\tilde{\mathbf{A}})\|_F
\end{aligned}$$

Applying the Lipschitz concentration for Gaussian measure (Ledoux and Talagrand, 2013) 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. ■

11.3 Proofs for the results in Section 4

Proof of Theorem 8 The equivalence of weight decay and ℓ_1 regularized problems in the theorem statement follows from the scaling argument as in the proofs of Lemma 1 and 2. The rest of the proof follows a similar argument as in the proof of Theorem 1. We reparameterize (16) as follows

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

Then taking the convex dual of the above problem with respect to the second layer weights yields the claimed form. ■

Proof of Theorem 9 Let us first restate the dual problem as

$$\max_{\mathbf{v}} -\frac{1}{2}\|\mathbf{v} - \mathbf{y}\|_2^2 + \frac{1}{2}\|\mathbf{y}\|_2^2 \text{ s.t. } \max_{\mathbf{u} \in \mathcal{B}_2} |\mathbf{v}^T \big(\mathbf{A}\mathbf{u}\big)_+| \leq \beta. \tag{58}$$

Since $\mathbf{A}$ is whitened such that $\mathbf{A}\mathbf{A}^T = \mathbf{I}_n$, (58) can be rewritten as follows

$$\max_{\mathbf{v}} -\frac{1}{2}\|\mathbf{v} - \mathbf{y}\|_2^2 + \frac{1}{2}\|\mathbf{y}\|_2^2 \text{ s.t. } \max\Big\{\|\big(\mathbf{v}\big)_+\|_2, \|\big(-\mathbf{v}\big)_+\|_2\Big\} \leq \beta.$$

The problem above has a closed-form solution in the following form

$$\mathbf{v}^* = \begin{cases} \beta\frac{\big(\mathbf{y}\big)_+}{\|\big(\mathbf{y}\big)_+\|_2} - \beta\frac{\big(-\mathbf{y}\big)_+}{\|\big(-\mathbf{y}\big)_+\|_2} & \text{if } \beta \leq \|\big(\mathbf{y}\big)_+\|_2, \ \beta \leq \|\big(-\mathbf{y}\big)_+\|_2 \\ \big(\mathbf{y}\big)_+ - \beta\frac{\big(-\mathbf{y}\big)_+}{\|\big(-\mathbf{y}\big)_+\|_2} & \text{if } \beta > \|\big(\mathbf{y}\big)_+\|_2, \ \beta \leq \big(-\mathbf{y}\big)_+\|_2 \\ \beta\frac{\big(\mathbf{y}\big)_+}{\|\big(\mathbf{y}\big)_+\|_2} - \big(-\mathbf{y}\big)_+ & \text{if } \beta \leq \|\big(\mathbf{y}\big)_+\|_2, \ \beta > \|\big(-\mathbf{y}\big)_+\|_2 \\ \mathbf{y} & \text{if } \beta > \|\big(\mathbf{y}\big)_+\|_2, \ \beta > \|\big(-\mathbf{y}\big)_+\|_2 \end{cases}$$

and the corresponding extreme points for the two-sided constraint in (58) are

$$\mathbf{U}^* = \begin{cases} \left(\frac{\mathbf{A}^\dagger\big(\mathbf{y}\big)_+}{\|\mathbf{A}^\dagger\big(\mathbf{y}\big)_+\|_2}, \frac{\mathbf{A}^\dagger\big(-\mathbf{y}\big)_+}{\|\mathbf{A}^\dagger\big(-\mathbf{y}\big)_+\|_2}\right) & \text{if } \beta \leq \|\big(\mathbf{y}\big)_+\|_2, \ \beta \leq \|\big(-\mathbf{y}\big)_+\|_2 \\ \frac{\mathbf{A}^\dagger\big(-\mathbf{y}\big)_+}{\|\mathbf{A}^\dagger\big(-\mathbf{y}\big)_+\|_2} & \text{if } \beta > \|\big(\mathbf{y}\big)_+\|_2, \ \beta \leq \big(-\mathbf{y}\big)_+\|_2 \\ \frac{\mathbf{A}^\dagger\big(\mathbf{y}\big)_+}{\|\mathbf{A}^\dagger\big(\mathbf{y}\big)_+\|_2} & \text{if } \beta \leq \|\big(\mathbf{y}\big)_+\|_2, \ \beta > \|\big(\mathbf{y}\big)_+\|_2 \\ \mathbf{0} & \text{if } \beta > \|\big(\mathbf{y}\big)_+\|_2, \ \beta > \|\big(-\mathbf{y}\big)_+\|_2 \end{cases}.$$

Now, we first substitute each case into the primal problem 16 and then take the derivative with respect to the output layer weights $\mathbf{w}$. Since this is a linear unconstrained least squares optimization problem, which is convex, we obtain the following closed-form solutions for the output layer weights

$$\mathbf{w}^* = \begin{cases} \begin{bmatrix} \|\mathbf{A}^\dagger \left(\mathbf{y}\right)_+\|_2 - \beta \\ -\|\mathbf{A}^\dagger \left(-\mathbf{y}\right)_+\|_2 + \beta \end{bmatrix} & \text{if } \beta \leq \|\left(\mathbf{y}\right)_+\|_2,\ \beta \leq \|\left(-\mathbf{y}\right)_+\|_2 \\ -\|\mathbf{A}^\dagger \left(-\mathbf{y}\right)_+\|_2 + \beta & \text{if } \beta > \|\left(\mathbf{y}\right)_+\|_2,\ \beta \leq \|\left(-\mathbf{y}\right)_+\|_2 \\ \|\mathbf{A}^\dagger \left(\mathbf{y}\right)_+\|_2 - \beta & \text{if } \beta \leq \|\left(\mathbf{y}\right)_+\|_2,\ \beta > \|\left(-\mathbf{y}\right)_+\|_2 \\ 0 & \text{if } \beta > \|\left(\mathbf{y}\right)_+\|_2,\ \beta > \|\left(-\mathbf{y}\right)_+\|_2 \end{cases}.$$

■

Proof of Theorem 10 The proof follows from a similar argument as in the proof of Theorem 1 and 8. We first put (18) into the following form

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

Then taking the dual of the above problem yields the claimed form. ■

Proof of Theorem 12 Since $\mathbf{A}$ is whitened, as a direct consequence of Theorem 9 and 10, we rewrite the dual problem as

$$\max_{\mathbf{v}} \mathbf{v}^T \mathbf{y} \text{ s.t. } \max \|\left(\mathbf{v}\right)_+\|_2, \|\left(-\mathbf{v}\right)_+\|_2 \leq \beta, \tag{59}$$

which has the following optimal solution

$$\mathbf{v}^* = \beta \frac{\left(\mathbf{y}\right)_+}{\|\left(\mathbf{y}\right)_+\|_2} - \beta \frac{\left(-\mathbf{y}\right)_+}{\|\left(-\mathbf{y}\right)_+\|_2}$$

and the corresponding extreme points are

$$\mathbf{U}^* = \begin{bmatrix} \frac{\mathbf{A}^\dagger \left(\mathbf{y}\right)_+}{\|\mathbf{A}^\dagger \left(\mathbf{y}\right)_+\|_2} & \frac{\mathbf{A}^\dagger \left(-\mathbf{y}\right)_+}{\|\mathbf{A}^\dagger \left(-\mathbf{y}\right)_+\|_2} \end{bmatrix} = \begin{bmatrix} \frac{\mathbf{A}^\dagger \left(\mathbf{y}\right)_+}{\sqrt{n_+}} & \frac{\mathbf{A}^\dagger \left(-\mathbf{y}\right)_+}{\sqrt{n_-}} \end{bmatrix}$$

where n_+ and n_- are the number of samples with positive and negative labels, respectively and the second equality follows from $y_i \in \{\pm 1\}$. If we substitute $\mathbf{U}^*$ into (18) and take derivative of the objective with respect to the output layer weight $\mathbf{w}$, we obtain the following optimality conditions

$$\begin{aligned} 0 &\in \partial \max\left\{0, \sqrt{n_+} - w_1\right\} \sqrt{n_+} + \beta \partial |w_1| \\ 0 &\in \partial \max\left\{0, \sqrt{n_-} + w_2\right\} \sqrt{n_-} + \beta \partial |w_2| \end{aligned}. \tag{60}$$

Therefore, the optimal solutions are

$$w_1 = \begin{cases} w \in [0, \sqrt{n_+}] & \text{if } \beta = \sqrt{n_+} \\ \sqrt{n_+} & \text{if } \beta < \sqrt{n_+} \\ 0 & \text{if } \beta > \sqrt{n_+} \end{cases}, \quad w_2 = \begin{cases} w \in [-\sqrt{n_-}, 0] & \text{if } \beta = \sqrt{n_-} \\ -\sqrt{n_-} & \text{if } \beta < \sqrt{n_-} \\ 0 & \text{if } \beta > \sqrt{n_-} \end{cases}.$$

■

Proof of Theorem 13 The proof follows from classical Fenchel duality (Boyd and Vandenberghe, 2004), and a similar argument as in the proof of Theorem 1 and 8. We first describe (19) in an equivalent form as follows

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

Then the dual function is

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

Therefore, using the classical Fenchel duality (Boyd and Vandenberghe, 2004) yields the proposed dual form. ■

Proof of Lemma 11 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 j^{th} column and row of $\mathbf{U}$ and $\mathbf{W}$, respectively. Then, (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

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

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 $\sum_{j=1}^m \|\mathbf{w}_j\|_2$. ■

Proof of Corollary 11 The proof directly follows from the proof of Corollary 3. ■

Proof of Corollary 12 The proof directly follows from the proof of Corollary 5. ■

Proof of Theorem 14 Proof directly follows from Corollary 11 and 12. ■

Proof of Theorem 15 We first apply the scaling in Lemma 11 and then restate the dual problem as

$$\max_{\mathbf{V}} -\frac{1}{2}\|\mathbf{V}-\mathbf{Y}\|_F^2+\frac{1}{2}\|\mathbf{Y}\|_F^2 \text{ s.t. } \max_{\mathbf{u}\in\mathcal{B}_2}\|\mathbf{V}^T\left(\mathbf{A}\mathbf{u}\right)_+\|_2\leq\beta. \tag{61}$$

Since $\mathbf{A}$ is whitened such that $\mathbf{A}\mathbf{A}^T=\mathbf{I}_n$ and $\mathbf{Y}$ is one hot encoded, (61) can be rewritten as follows

$$D:=\max_{\mathbf{V}} -\frac{1}{2}\|\mathbf{V}-\mathbf{Y}\|_F^2+\frac{1}{2}\|\mathbf{Y}\|_F^2 \text{ s.t. } \max_{\mathbf{u}\in\mathcal{B}_2}\|\mathbf{V}^T\mathbf{u}\|_2\leq\beta. \tag{62}$$

The problem above has a closed-form solution as follows

$$\mathbf{v}_j^*=\begin{cases}\beta\frac{\mathbf{y}_j}{\|\mathbf{y}_j\|_2} & \text{if } \beta\leq\|\mathbf{y}_j\|_2\\ \mathbf{y}_j & \text{otherwise}\end{cases},\quad \forall j\in[o]. \tag{63}$$

and the corresponding extreme points of the constraint in (62) are

$$\mathbf{u}_j^*=\begin{cases}\frac{\mathbf{A}^\dagger\mathbf{y}_j}{\|\mathbf{A}^\dagger\mathbf{y_j}\|_2} & \text{if } \beta\leq\|\mathbf{y}_j\|_2\\ - & \text{otherwise}\end{cases},\quad \forall j\in[o]. \tag{64}$$

Now let us first denote the set of indices that yield an extreme point as $\mathcal{E}:=\{j:\beta\leq\|\mathbf{y}_j\|_2, j\in[o]\}$. Then we compute the objective value for the dual problem in (62) using the optimal parameter in (63)

$$\begin{aligned}D&=-\frac{1}{2}\|\mathbf{V}^*-\mathbf{Y}\|_F^2+\frac{1}{2}\|\mathbf{Y}\|_F^2\\&=-\frac{1}{2}\sum_{j\in\mathcal{E}}(\beta-\|\mathbf{y}_j\|_2)^2+\frac{1}{2}\sum_{j=1}^{o}\|\mathbf{y}_j\|_2^2\\&=-\frac{1}{2}\beta^2|\mathcal{E}|+\beta\sum_{j\in\mathcal{E}}\|\mathbf{y}_j\|_2+\frac{1}{2}\sum_{j\notin\mathcal{E}}\|\mathbf{y}_j\|_2^2.\end{aligned} \tag{65}$$

Next, we restate the primal problem as follows

$$P:=\max_{\mathbf{U},\mathbf{W}}\frac{1}{2}\|\left(\mathbf{A}\mathbf{U}\right)_+\mathbf{W}-\mathbf{Y}\|_F^2+\beta\sum_{j=1}^{o}\|\mathbf{w}_j\|_2 \text{ s.t. } \|\mathbf{u}_j\|\leq 1,\forall j\in[o], \tag{66}$$

and then solve it using the optimal hidden layer weights in (64), which yields the following optimal solution

$$\left(\mathbf{u}_j^*,\mathbf{w}_j^*\right)=\begin{cases}\left(\frac{\mathbf{A}^\dagger\mathbf{y}_j}{\|\mathbf{A}^\dagger\mathbf{y_j}\|_2},\left(\|\mathbf{y}_j\|_2-\beta\right)\mathbf{e}_j\right) & \text{if } \beta\leq\|\mathbf{y}_j\|_2\\ - & \text{otherwise}\end{cases},\quad \forall j\in[o]. \tag{67}$$

Evaluating the primal problem objective with the parameters (67) gives

$$\begin{aligned} P &= \frac{1}{2}\|\left(\mathbf{A}\mathbf{U}^*\right)_+\mathbf{W}^* - \mathbf{Y}\|_F^2 + \beta\sum_{j\in\mathcal{E}}\|\mathbf{w}_j^*\|_2 \\ &= \frac{1}{2}\left\|\sum_{j\in\mathcal{E}}\left(\|\mathbf{y}_j\|_2-\beta\right)\frac{\mathbf{y}_j}{\|\mathbf{y}_j\|_2}\mathbf{e}_j^T - \mathbf{Y}\right\|_F^2 + \beta\sum_{j\in\mathcal{E}}(\|\mathbf{y}_j\|_2-\beta) \\ &= \frac{1}{2}\sum_{j\in\mathcal{E}}\left\|\beta\frac{\mathbf{y}_j}{\|\mathbf{y}_j\|_2}\mathbf{e}_j^T\right\|_F^2 + \frac{1}{2}\sum_{j\notin\mathcal{E}}\|\mathbf{y}_j\mathbf{e}_j^T\|_F^2 + \beta\sum_{j\in\mathcal{E}}\|\mathbf{y}_j\|_2 - \beta^2|\mathcal{E}| \\ &= -\frac{1}{2}\beta^2|\mathcal{E}| + \frac{1}{2}\sum_{j\notin\mathcal{E}}\|\mathbf{y}_j\|_2^2 + \beta\sum_{j\in\mathcal{E}}\|\mathbf{y}_j\|_2, \end{aligned} \tag{68}$$

which has the same value with (65). Therefore, strong duality holds, i.e., $P = D$, and the set of weight in (67) are optimal for the primal problem (66). We also note that since $\mathbf{Y}$ is one hot encoded, (67) can be equivalently stated as

$$\left(\mathbf{u}_j^*, \mathbf{w}_j^*\right) = \begin{cases} \left(\frac{\mathbf{A}^\dagger\mathbf{y}_j}{\|\mathbf{A}^\dagger\mathbf{y_j}\|_2}, \left(\sqrt{n_j}-\beta\right)\mathbf{e}_j\right) & \text{if } \beta \le \sqrt{n_j} \\ - & \text{otherwise} \end{cases}, \quad \forall j \in [o], \tag{69}$$

where n_j is the number samples in the j^{th} class. ■

Proof of Lemma 12 The proof is a straightforward generalization of the scalar output case in Lemma 9. ■

Proof of Lemma 13 The proof is a straightforward generalization of the scalar output case in Lemma 10. ■

12. 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. Rockafellar (1970)). 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 (12) as

$$D^* = \inf_{t \in \mathbb{R}} t \quad \text{s.t. } \mathbf{y} \in t \operatorname{conv}\left\{\mathcal{Q}_{\mathbf{A}} \cup -\mathcal{Q}_{\mathbf{A}}\right\}, \tag{70}$$

where conv represents the convex hull of a set.

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

$$\max_{\mathbf{v}} \mathbf{v}^T \mathbf{y} \text{ s.t. } \mathbf{v} \in \mathcal{Q}_{\mathbf{A}}^{\circ} \ , \ -\mathbf{v} \in \mathcal{Q}_{\mathbf{A}}^{\circ} \tag{71}$$

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

Remark 2 *The dual problem given in* (71) *is analogous to the convex duality in minimum ℓ_1-norm solutions to linear systems. In particular, for the latter it holds that*

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

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.

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

which parallels the gauge optimization form in (70).

References

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

Sanjeev Arora, Nadav Cohen, and Elad Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. In *35th International Conference on Machine Learning, ICML 2018*, pages 372–389. International Machine Learning Society (IMLS), 2018.

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. URL `http://arxiv.org/abs/1904.11955`.

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

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.

Burak Bartan and Mert Pilanci. Convex relaxations of convolutional neural nets. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4928–4932. IEEE, 2019.

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.

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

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. URL `http://arxiv.org/abs/1904.09080`.

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

Alon Brutzkus, Amir Globerson, Eran Malach, and Shai Shalev-Shwartz. SGD learns over-parameterized networks that provably generalize on linearly separable data. *CoRR*, abs/1710.10174, 2017. URL `http://arxiv.org/abs/1710.10174`.

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

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

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

Chris HQ Ding, Tao Li, and Michael I Jordan. Convex and semi-nonnegative matrix factorizations. *IEEE transactions on pattern analysis and machine intelligence*, 32(1):45–55, 2008.

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.

Simon S Du and Jason D Lee. On the power of over-parametrization in neural networks with quadratic activation. *arXiv preprint arXiv:1803.01206*, 2018.

Simon S. Du, Xiyu Zhai, Barnabás Póczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. *CoRR*, abs/1810.02054, 2018. URL `http://arxiv.org/abs/1810.02054`.

Tolga Ergen and Mert Pilanci. Convex optimization for shallow neural networks. In *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 79–83. IEEE, 2019a.

Tolga Ergen and Mert Pilanci. Convex duality and cutting plane methods for over-parameterized neural networks. In *OPT-ML workshop*, 2019b.

Tolga Ergen and Mert Pilanci. Convex geometry of two-layer relu networks: Implicit autoencoding and interpretable models. In *International Conference on Artificial Intelligence and Statistics*, pages 4024–4033. PMLR, 2020a.

Tolga Ergen and Mert Pilanci. Revealing the structure of deep neural networks via convex duality. 2020b.

Tolga Ergen and Mert Pilanci. Convex programs for global optimization of convolutional neural networks in polynomial-time. In *OPT-ML workshop*, 2020c.

Tolga Ergen and Mert Pilanci. Implicit convex regularizers of {cnn} architectures: Convex optimization of two- and three-layer networks in polynomial time. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=0N8jUH4JMv6`.

Tolga Ergen, Arda Sahiner, Batu Ozturkler, John M. Pauly, Morteza Mardani, and Mert Pilanci. Demystifying batch normalization in relu networks: Equivalent convex optimization models and implicit regularization. *CoRR*, abs/2103.01499, 2021. URL `https://arxiv.org/abs/2103.01499`.

Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2):95–110, 1956. doi: 10.1002/nav.3800030109. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800030109`.

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.

Miguel Angel Goberna and Marco López-Cerdá. *Linear semi-infinite optimization*. 01 1998. doi: 10.1007/978-1-4899-8044-1_3.

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.

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

Vikul Gupta, Burak Bartan, Tolga Ergen, and Mert Pilanci. Convex neural autoregressive models: Towards tractable, expressive, and theoretically-backed models for sequential forecasting and generation. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3890–3894, 2021. doi: 10.1109/ICASSP39728.2021.9413662.

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

Lei Huang, Dawei Yang, Bo Lang, and Jia Deng. Decorrelated batch normalization. 2018.

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.

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

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

Jonathan Lacotte and Mert Pilanci. All local minima are global for two-layer relu neural networks: The hidden convex optimization landscape. *arXiv preprint arXiv:2006.05900*, 2020.

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

Johannes Lederer. No spurious local minima: on the optimization landscapes of wide and deep neural networks, 2020.

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

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.

Yuanzhi Li and Yingyu Liang. Learning overparameterized neural networks via stochastic gradient descent on structured data. *CoRR*, abs/1808.01204, 2018. URL `http://arxiv.org/abs/1808.01204`.

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

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.

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

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

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.

Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. Towards understanding the role of over-parametrization in generalization of neural networks. *arXiv preprint arXiv:1805.12076*, 2018.

Greg Ongie, Rebecca Willett, Daniel Soudry, and Nathan Srebro. A function space view of bounded norm infinite width relu nets: The multivariate case. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=H1lNPxHKDH`.

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

Mert Pilanci and Tolga Ergen. Neural networks are convex regularizers: Exact polynomial-time convex optimization formulations for two-layer networks. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7695–7705. PMLR, 13–18 Jul 2020. URL http://proceedings.mlr.press/v119/pilanci20a.html.

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

Saharon Rosset, Grzegorz Swirszcz, Nathan Srebro, and Ji Zhu. L1 regularization in infinite dimensional feature spaces. In *International Conference on Computational Learning Theory*, pages 544–558. Springer, 2007.

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

Arda Sahiner, Tolga Ergen, Batu Ozturkler, Burak Bartan, John Pauly, Morteza Mardani, and Mert Pilanci. Hidden convexity of wasserstein gans: Interpretable generative models with closed-form solutions, 2021a.

Arda Sahiner, Tolga Ergen, John M. Pauly, and Mert Pilanci. Vector-output relu neural network problems are copositive programs: Convex analysis of two layer networks and polynomial-time algorithms. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=fGF8qAqpXXG.

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. URL http://arxiv.org/abs/1902.05040.

Vaishaal Shankar, Alex Fang, Wenshuo Guo, Sara Fridovich-Keil, Jonathan Ragan-Kelley, Ludwig Schmidt, and Benjamin Recht. Neural kernels without tangents. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8614–8623. PMLR, 13–18 Jul 2020. URL http://proceedings.mlr.press/v119/shankar20a.html.

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

Louis Thiry, Michael Arbel, Eugene Belilovsky, and Edouard Oyallon. The unreasonable effectiveness of patches in deep convolutional kernels methods. 2021.

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

RH Tütüncü, KC Toh, and MJ Todd. Sdpt3—a matlab software package for semidefinite-quadratic-linear programming, version 3.0. *Web page http://www. math. nus. edu. sg/mattohkc/sdpt3. html*, 2001.

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.

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

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.

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

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