Neural Networks are Convex Regularizers: Exact Polynomial-time Convex Optimization Formulations for Two-Layer Networks

Mert Pilanci [1] **Tolga Ergen** [1]

Abstract

We develop exact representations of two layer neural networks with rectified linear units in terms of a single convex program with number of variables polynomial in the number of training samples and number of hidden neurons. Our theory utilizes semi-infinite duality and minimum norm regularization. Moreover, we show that certain standard multi-layer convolutional neural networks are equivalent to L1 regularized linear models in a polynomial sized discrete Fourier feature space. We also introduce exact semi-definite programming representations of convolutional and fully connected linear multi-layer networks which are polynomial size in both the sample size and dimension.

1. Introduction

In this paper, we introduce a finite dimensional, polynomial-size convex program that globally solves the training problem for two-layer neural networks with rectified linear unit activation functions. The key to our analysis is a generic convex duality method we introduce, and is of independent interest for other non-convex problems. We further prove that strong duality holds in a variety of architectures.

1.1. Related work and overview

Convex neural network training was considered in the literature (Bengio et al., 2006; Bach, 2017). However, convexity arguments in the existing work are restricted to infinite width networks, where infinite dimensional optimization problems need to be solved. In fact, adding even a single neuron to the model requires the solution of a non-convex problem where no efficient algorithm is known (Bach, 2017). In this work, we develop a novel duality theory and introduce polynomial-time finite dimensional convex programs, which are exact and computationally tractable.

[1]Department of Electrical Engineering, Stanford University, CA, USA. Correspondence to: Mert Pilanci <pilanci@stanford.edu>, Tolga Ergen <ergen@stanford.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 108, 2020.

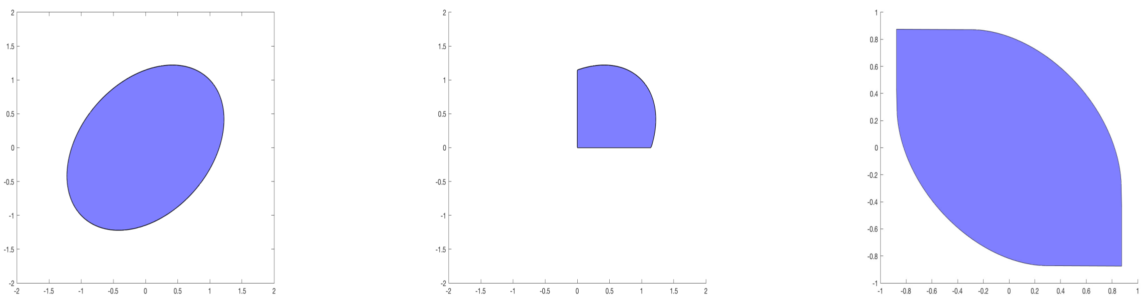
Several recent studies considered over-parameterized neural networks, where the width approaches infinity by leveraging connections to kernel methods, and showed that randomly initialized gradient descent can fit all the training samples (Jacot et al., 2018; Du et al., 2019; Allen-Zhu et al., 2019). However, in this *kernel regime*, the analysis shows that almost no hidden neurons move from their initial values to actively learn useful features (Chizat & Bach, 2018). Experiments also confirm that the kernel approximation as the width tends to infinity is unable to fully explain the success of non-convex neural network models (Arora et al., 2019). On the contrary, our work precisely characterizes the mechanism behind extraordinary properties of neural network models for any finite number of hidden neurons. We prove that networks with rectified linear units are identical to *convex regularization* methods in a finite higher dimensional space.

Consider a two-layer neural network $f : \mathbb{R}^d \to \mathbb{R}$ with m hidden neurons and a scalar output.

$$f(x) = \sum_{j=1}^{m} \phi(x^T u_j)\alpha_j \,, \tag{1}$$

where $u_1, \ldots, u_m \in \mathbb{R}^d$ are the weight vectors of hidden neurons, $\alpha_1, \ldots, \alpha_m \in \mathbb{R}$ are the corresponding second layer weights, and ϕ is the activation function. We will assume that $\phi(t) = (t)_+ := \max(t, 0)$ is the ReLU activation function throughout the paper unless noted otherwise. We extend the definition of scalar functions to vectors and matrices entry-wise. We use $\mathcal{B}_p$ to denote the unit ℓ_p ball in $\mathbb{R}^d$. We denote the set of integers from 1 to n as $[n]$. Furthermore, we use σ to denote singular values.

In order to keep the notation simple and clearly convey the main idea, we will restrict our attention to two-layer ReLU networks with scalar output trained with squared loss. All of our results immediately extend to vector outputs, tensor inputs, arbitrary convex classification and regression loss functions, and certain other network architectures (see Appendix).



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

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

(c) Polar set $(\mathcal{Q}_X \cup -\mathcal{Q}_X)^\circ$: $\{v \,|\, |v^T w| \leq 1 \, \forall w \in \mathcal{Q}_X\}$

Figure 1: Sets involved in the construction of the Neural Gauge. Ellipsoidal set, rectified ellipsoid $\mathcal{Q}_X$ and the polar of $\mathcal{Q}_X \cup -\mathcal{Q}_X$.

Given a data matrix $X \in \mathbb{R}^{n \times d}$ and label vector $y \in \mathbb{R}^n$, and a regularization parameter $\beta > 0$, consider minimizing the sum of the squared loss objective and squared ℓ_2-norm of all parameters

$$p^* := \min_{\{\alpha_j, u_j\}_{j=1}^m} \frac{1}{2}\Big\| \sum_{j=1}^m (Xu_j)_+ \alpha_j - y \Big\|_2^2 + \frac{\beta}{2} \sum_{j=1}^m (\|u_j\|_2^2 + \alpha_j^2) . \tag{2}$$

The above objective is highly non-convex due to non-linear ReLU activations and product between hidden and outer layer weights. The best known algorithm for globally minimizing the above objective is a brute-force search over all possible piece-wise linear regions of ReLU activations of m neurons and output layer sign patterns (Arora et al., 2018). This algorithm has complexity $O(2^m n^{dm})$ (see Theorem 4.1 in (Arora et al., 2018)). In fact, known algorithms for approximately learning m hidden neuron ReLU networks have complexity $O(2^{\sqrt{m}})$ (see Theorem 5 of (Goel et al., 2017)) due to similar combinatorial explosion with the number of neurons m.

2. Convex duality for two-layer networks

Now we introduce our main technical tool for deriving convex representations of the non-convex objective function (2). We start with the ℓ_1 penalized representation, which is equivalent to (2) (see Appendix).

$$p^* = \min_{\substack{\|u_j\|_2 \leq 1 \\ \forall j \in [m]}} \min_{\{\alpha_j\}_{j=1}^m} \frac{1}{2}\Big\| \sum_{j=1}^m (Xu_j)_+ \alpha_j - y \Big\|_2^2 + \beta \sum_{j=1}^m |\alpha_j| , \tag{3}$$

Replacing the inner minimization problem with its convex dual, we obtain (see Appendix)

$$p^* = \min_{\substack{\|u_j\|_2 \leq 1 \\ \forall j \in [m]}} \max_{\substack{v \in \mathbb{R}^n \text{ s.t.} \\ |v^T (Xu_j)_+| \leq \beta, \forall j \in [m]}} -\frac{1}{2}\|y - v\|_2^2 + \frac{1}{2}\|y\|_2^2 .$$

Interchanging the order of $\min$ and $\max$, we obtain the lower-bound d^* via weak duality

$$p^* \geq d^* := \max_{\substack{v \in \mathbb{R}^n \text{ s.t.} \\ |v^T (Xu)_+| \leq \beta \, \forall u \in \mathcal{B}_2}} -\frac{1}{2}\|y - v\|_2^2 + \frac{1}{2}\|y\|_2^2 , \tag{4}$$

where $\mathcal{B}_2$ is the unit ℓ_2 ball in $\mathbb{R}^d$. The above problem is a convex *semi-infinite* optimization problem with n variables and infinitely many constraints. We will show that strong duality holds, i.e., $p^* = d^*$ as long as the number of hidden neurons m satisfies $m \geq m^*$ for some $m^* \in \mathbb{N}$, $1 \leq m^* \leq n$, which will be defined in the sequel. As it will be shown, m^* can be smaller than n. The dual of the dual program (4) can be derived using standard semi-infinite programming theory (Goberna & López-Cerdá, 1998), and corresponds to the bi-dual of the non-convex problem (2).

Now we briefly introduce basic properties of signed measures that is necessary to state the dual of (4) and refer the reader to (Rosset et al., 2007; Bach, 2017) for further details. Consider an arbitrary measurable input space $\mathcal{X}$ with a set of continuous basis functions $\phi_u : \mathcal{X} \to \mathbb{R}$ parameterized by $u \in \mathcal{B}_2$. We then consider real-valued Radon measures equipped with the uniform norm (Rudin, 1964). For a signed Radon measure μ, we can define an infinite width neural network output for the input $x \in \mathcal{X}$ as $f(x) = \int_{u \in \mathcal{B}_2} \phi_u(x) d\mu(u)$. The total variation norm of the signed measure μ is defined as the supremum of $\int_{u \in \mathcal{B}_2} q(u) d\mu(u)$ over all continuous functions $q(u)$ that satisfy $|q(u)| \leq 1$. Consider the ReLU basis functions $\phi_u(x) = \left(x^T u\right)_+$. We may express networks with finitely many neurons as in (1) by

$$f(x) = \sum_{j=1}^m \phi_{u_j}(x) \alpha_j ,$$

which corresponds to $\mu = \sum_{j=1}^m \alpha_j \delta(u - u_j)$ where δ is the Dirac delta function. And the total variation norm $\|\mu\|_{TV}$ of μ reduces to the ℓ_1 norm $\|\alpha\|_1$.

Now we can state the dual of (4) (see Section 2 of (Shapiro, 2009) and Section 8.6 of (Goberna & López-Cerdá, 1998)) as follows

$$d^* \leq p^*_\infty = \min_\mu \frac{1}{2}\left\|\int_{u\in\mathcal{B}_2} \left(Xu\right)_+ d\mu(u) - y\right\|_2^2 + \beta\left\|\mu\right\|_{TV} \quad (5)$$

where $\|\mu\|_{TV}$ stands for the total variation norm of the Radon measure μ. Furthermore, an application of Caratheodory's theorem shows that the infinite dimensional bi-dual (5) always has a solution that is supported with finitely many Dirac delta functions, whose exact number we define as m^*, where $m^* \leq n+1$ (Rosset et al., 2007). Therefore we have

$$\begin{aligned} p_\infty * = &\min_{\substack{\|u_j\|_2\leq 1\\ \{\alpha_j,u_j\}_{j=1}^{m^*}}} \frac{1}{2}\Big\|\sum_{j=1}^{m^*}(Xu_j)_+\alpha_j - y\Big\|_2^2 + \beta\sum_{j=1}^{m^*}|\alpha_j|\,, \\ &= p^*\,, \end{aligned}$$

as long as $m \geq m^*$. We show that strong duality holds, i.e., $d^* = p^*$ in Appendix A.8 and A.11. In the sequel, we illustrate how m^* can be determined via a finite-dimensional parameterization of (4) and its dual.

2.1. A geometric insight: Neural Gauge Function

An interesting geometric insight can be provided in the weakly regularized case where $\beta \to 0$. In this case, minimizers of (3) and hence (2) approach minimum norm interpolation $p^*_{\beta\to 0} := \lim_{\beta\to 0}\beta^{-1}p^*$ given by

$$\begin{aligned} p^*_{\beta\to 0} = &\min_{\{u_j,\alpha_j\}_{j=1}^m} \sum_{j=1}^m |\alpha_j| \\ &\text{s.t. } \sum_{j=1}^m (Xu_j)_+\alpha_j = y,\ \|u_j\|_2 \leq 1\,\forall j. \end{aligned} \quad (6)$$

It can be shown that $p^*_{\beta\to 0}$ is the gauge function of the convex hull of $\mathcal{Q}_X \cup -\mathcal{Q}_X$ where $\mathcal{Q}_X := \{(Xu)_+ : u \in \mathcal{B}_2\}$ (see Appendix), i.e.,

$$\begin{aligned} p^*_{\beta\to 0} = &\inf_{t:t\geq 0}\ t \\ &y \in t\,\mathrm{Conv}\{\mathcal{Q}_X \cup -\mathcal{Q}_X\}\,. \end{aligned}$$

We call the above problem *Neural Gauge* due to the connection to the minimum norm interpolation problem. Using classical polar gauge duality (see e.g. (Rockafellar, 1970), it holds that

$$\begin{aligned} p^*_{\beta\to 0} = &\max y^T z \\ &z \in (\mathcal{Q}_X \cup -\mathcal{Q}_X)^\circ\,, \end{aligned} \quad (7)$$

where $(\mathcal{Q}_X \cup -\mathcal{Q}_X)^\circ$ is the polar of the set $\mathcal{Q}_X \cup -\mathcal{Q}_X$. Therefore, evaluating the support function of this polar set is equivalent to solving the neural gauge problem, i.e., minimum norm interpolation $p^*_{\beta\to 0}$. These sets are illustrated in Figure 1. Note that the polar set $(\mathcal{Q}_X \cup -\mathcal{Q}_X)^\circ$ is always convex (see Figure 1c), which also appears in the dual problem (4) as a constraint. In particular, $\lim_{\beta\to 0}\beta^{-1}d^*$ is equal to the support function. Our finite dimensional convex program leverages the convexity and an efficient description of this set as we discuss next.

3. An exact finite dimensional convex program

Consider diagonal matrices $\mathrm{Diag}(1[Xu \geq 0])$ where $u \in \mathbb{R}^d$ is arbitrary and $1[Xu \geq 0] \in \{0,1\}^n$ is an indicator vector with Boolean elements $[1[x_1^T u \geq]0, \ldots, 1[x_n^T u \geq 0]]$. Let us enumerate all such distinct diagonal matrices that can be obtained for all possible $u \in \mathbb{R}^d$, and denote them as $D_1, ..., D_P$. P is the number of regions in a partition of $\mathbb{R}^d$ by hyperplanes passing through the origin, and are perpendicular to the rows of X. It is well known that

$$P \leq 2\sum_{k=0}^{r-1}\binom{n-1}{k} \leq 2r\Big(\frac{e(n-1)}{r}\Big)^r,$$

for $r \leq n$ where $r := \mathrm{rank}(X)$. (Ojha, 2000; Stanley et al., 2004; Winder, 1966; Cover, 1965) (see Appendix for details).

Consider the finite dimensional convex problem

$$\begin{aligned} &\min_{\{v_i,w_i\}_{i=1}^P} \frac{1}{2}\Big\|\sum_{i=1}^P D_i X(v_i - w_i) - y\Big\|_2^2 + \beta\sum_{i=1}^P(\|v_i\|_2 + \|w_i\|_2) \\ &\text{s.t. } (2D_i - I_n)Xv_i \geq 0,\ (2D_i - I_n)Xw_i \geq 0,\ \forall i \in [P]. \end{aligned} \quad (8)$$

Theorem 1. The convex program (8) and the non-convex problem (2) where $m \geq m^*$ has identical optimal values[1]. Moreover, an optimal solution to (2) with m^* neurons can be constructed from an optimal solution to (8) as follows

$$(u^*_{j_i}, \alpha^*_{j_i}) = \begin{cases} \left(\frac{v_i^*}{\sqrt{\|v_i^*\|_2}}, \sqrt{\|v_i^*\|_2}\right) & \text{if} \quad v_i^* \neq 0 \\ \left(\frac{w_i^*}{\sqrt{\|w_i^*\|_2}}, -\sqrt{\|w_i^*\|_2}\right) & \text{if} \quad w_i^* \neq 0\,, \end{cases}$$

where v_i^*, w_i^* are the optimal solutions to (8), and either v_i^* or w_i^* is non-zero for all $i = 1, ..., P$. We have $m^* = \sum_{j:v_j^*\neq 0 \text{ or } u_j^*\neq 0}^P 1$, where $\{v_i^*, w_i^*\}_{i=1}^P$ are optimal in (8).

Remark 3.1. *Theorem 1 shows that two layer ReLU networks with m hidden neurons can be globally optimized*

[1] m^* is defined as the number of Dirac deltas in the optimal solution to (5). If the optimum is not unique, we may pick the minimum cardinality solution.

via the second order cone program (8) *with* $2dP$ *variables and* $2nP$ *linear inequalities where* $P = 2r\left(\frac{e(n-1)}{r}\right)^r$, *and* $r = rank(X)$. *The computational complexity is* $O\left(r^3\left(\frac{n}{r}\right)^{3r}\right)$ *using standard interior-point solvers. For fixed rank* r *(or dimension* d*), the complexity is polynomial in* n *and* m*, which is an exponential improvement over the state of the art (Arora et al., 2018; Bienstock et al., 2018). Note that* d *is a small number that corresponds to the filter size in CNNs as we illustrate in the next section. However, for fixed* n *and* $rank(X) = d$*, the complexity is exponential in* d*, which can not be improved unless* $P = NP$ *even for* $m = 2$ *(Boob et al., 2018). We also remark that further theoretical insight as well as faster numerical solvers can be developed due to the similarity to group Lasso (Yuan & Lin, 2006) and related structured regularization methods.*

Remark 3.2. *We note that the convex program* (8) *can be approximated by sampling a set of diagonal matrices* $D_1, ..., D_{\tilde{P}}$*. For example, one can generate* $u \sim N(0, I_d)$*, or from any distribution* $\tilde{P}$ *times, and let* $D_1 = Diag(1[Xu_1 \geq 0]), ..., D_{\tilde{P}} = Diag(1[Xu_{\tilde{P}} \geq 0])$ *and solve the reduced convex problem, where remaining variables are set to zero. This is essentially a type of coordinate descent applied to* (8)*. In Section 6, we show that this approximation in fact works extremely well, often better than backpropagation. In fact, backpropagation (BP) can be viewed as a heuristic method to solve the convex objective* (8)*. The global optima of this convex program* (8) *are among the fixed points of BP, i.e., stationary points of* (2)*. Moreover, we can bound the suboptimality of any feasible solution, e.g., from backpropagation, in the non-convex cost* (2) *using the dual of* (8)*.*

The proof of Theorem 1 can be found in Section 5.

4. Convolutional neural networks

Here, we introduce extensions of our approach to convolutional neural networks (CNNs). Two-layer convolutional networks with m hidden neurons (filters) of dimension d and fully connected output layer weights (flattened activations) can be described by patch matrices $X_k \in \mathbb{R}^{n\times d}$, $k = 1, ..., K$ as $f(X_1, ..., X_K) = \sum_{j=1}^m \sum_{k=1}^K \phi(X_k u_j)\alpha_{jk}$. This formulation also includes image, or tensor inputs.

4.1. ReLU convolutional networks with vector outputs

Consider the training problem

$$\min_{\{\alpha_j, u_j\}_{j=1}^m} \frac{1}{2}\sum_{k=1}^K \Big\| \sum_{j=1}^m (X_k u_j)_+ \alpha_j - y_k \Big\|_2^2 + \frac{\beta}{2}\sum_{j=1}^m (\|u_j\|_2^2 + \alpha_j^2),$$

where $y_1, ..., y_K$ are known labels Then the above can be reduced to (2) by defining $X' = [X_1^T, ..., X_K^T]^T$ and $y' = [y_1, ..., y_K]^T$. Therefore, the convex program (8) solves the above problem exactly in $O\left(r^3\left(\frac{n}{r}\right)^{3r}\right)$ complexity, where r is the number of variables in a single filter. We note that typical CNNs use m filters of size 3×3 (r=9) in the first hidden layer (He et al., 2016).

4.2. Linear convolutional networks are Semi-definite Programs (SDPs)

We now start with the simple case of linear activations $\phi(t) = t$, where the training problem becomes

$$\min_{\{u_j, \alpha_j\}_{j=1}^m} \frac{1}{2}\Big\| \sum_{k=1}^K \sum_{j=1}^m X_k u_j \alpha_{jk} - y \Big\|_2^2 + \frac{\beta}{2}\sum_{j=1}^m \left(\|u_j\|_2^2 + \|\alpha_j\|_2^2\right). \tag{9}$$

The corresponding dual problem is given by

$$\max_v -\frac{1}{2}\|v - y\|_2^2 + \frac{1}{2}\|y\|_2^2 \text{ s.t. } \max_{\|u\|_2 \leq 1} \sum_k \left(v^T X_k u\right)^2 \leq 1. \tag{10}$$

Similar arguments to those used in the proof of Theorem 1, strong duality holds (see Appendix). Further, the maximizers of the inner problem are the maximal eigenvectors of $\sum_k X_k^T v v^T X_k$, which are optimal neurons (filters). We can express (10) as the SDP

$$\begin{aligned} &\max_v -\frac{1}{2}\|v - y\|_2^2 + \frac{1}{2}\|y\|_2^2 \\ &\text{s.t. } \sigma_{\max}\left([X_1^T v \ldots X_K^T v]\right) \leq \beta \end{aligned} \tag{11}$$

The dual of the above SDP is a nuclear norm penalized convex optimization problem (see Appendix)

$$\min_{z_k \in \mathbb{R}^d, \forall k} \frac{1}{2}\Big\| \sum_{k=1}^K X_k z_k - y \Big\|_2^2 + \beta \Big\| [z_1, \ldots, z_K] \Big\|_*, \tag{12}$$

where $\Big\|[z_1, ..., z_K]\Big\|_* = \|Z\|_* := \sum_i \sigma_i(Z)$ is the ℓ_1 norm of singular values, i.e., nuclear norm (Recht et al., 2010).

4.3. Linear circular convolutional networks

Now, if we assume that the patches are padded with enough zeros and extracted with stride one, then the circular version of (9) can be written as

$$\min_{\{u_j, \alpha_j\}_{j=1}^m} \frac{1}{2}\Big\| \sum_{j=1}^m X U_j \alpha_j - y \Big\|_2^2 + \frac{\beta}{2}\sum_{j=1}^m \left(\|u_j\|_2^2 + \|\alpha_j\|_2^2\right) \tag{13}$$

where $U_j \in \mathbb{R}^{d\times d}$ is a circulant matrix generated by a circular shift modulo d using the elements $u_i \in \mathbb{R}^h$. Then, the SDP (11) reduces to (see Appendix)

$$\min_{z\in\mathbb{C}^d} \frac{1}{2}\left\|\tilde{X}z - y\right\|_2^2 + \beta\|z\|_1, \tag{14}$$

where $\tilde{X} = XF$, and $F \in \mathbb{C}^{d\times d}$ is the DFT matrix.

5. Proof of the main result (Theorem 1)

We now prove the main result for scalar output two-layer ReLU networks with squared loss[2]. We start with the dual representation

$$\begin{aligned} &\max_{v} -\frac{1}{2}\|v-y\|_2^2 + \frac{1}{2}\|y\|_2^2 \\ &\text{s.t.} \max_{u:\,\|u\|_2\le 1} |v^T(Xu)_+| \le \beta\,. \end{aligned} \tag{15}$$

Note that the constraint (15) can be represented as

$$\big\{v : \max_{\|u\|_2\le 1} v^T(Xu)_+ \le \beta\big\} \cap \big\{v : \max_{\|u\|_2\le 1} -v^T(Xu)_+ \le \beta\big\}.$$

We now focus on a single-sided dual constraint

$$\max_{u:\,\|u\|_2\le 1} v^T(Xu)_+ \le \beta, \tag{16}$$

by considering hyperplane arrangements and a convex duality argument over each partition. We first partition $\mathbb{R}^d$ into the following subsets

$$P_S := \{u \,:\, x_i^T u \ge 0, \forall i \in S,\ x_j^T u \le 0, \forall j \in S^c\}$$

Let $\mathcal{H}_X$ be the set of all hyperplane arrangement patterns for the matrix X, defined as the following set

$$\mathcal{H}_X = \bigcup \big\{\{\text{sign}(Xu)\} \,:\, u \in \mathbb{R}^d\big\}\,.$$

It is obvious that the set $\mathcal{H}$ is bounded, i.e., $\exists N_H \in \mathbb{N} < \infty$ such that $|\mathcal{H}| \le N_H$.

We next define an alternative representation of the sign patterns in $\mathcal{H}_X$, which is the collection of sets that correspond to positive signs for each element in $\mathcal{H}$. More precisely, let

$$\mathcal{S}_X := \big\{\{\cup_{h_i=1}\{i\}\} \,:\, h \in \mathcal{H}_X\big\}$$

We now express the maximization in the dual constraint in (16) over all possible hyperplane arrangement patterns as

$$\begin{aligned} &\max_{u:\,\|u\|_2\le 1} v^T(Xu)_+ \\ &= \max_{u:\,\|u\|_2\le 1} v^T\text{Diag}(Xu\ge 0)Xu \\ &= \max_{\substack{S\subseteq\{1,\dots,n\}\\ S\in\mathcal{S}_X}} \max_{\substack{u:\,\|u\|_2\le 1\\ x_i^Tu\ge 0\ \forall i\in S\\ x_j^Tu\le 0\ \forall j\in S^c}} v^T\text{Diag}(Xu\ge 0)Xu \\ &= \max_{\substack{S\subseteq\{1,\dots,n\}\\ S\in\mathcal{S}_X}} \max_{\substack{u:\,\|u\|_2\le 1\\ u\in P_S}} v^T\text{Diag}(Xu\ge 0)Xu \end{aligned}$$

Let us define the diagonal matrix $D(S) \in \mathbb{R}^{n\times n}$ which is a function of the subset $S \subseteq \{1,\dots,n\}$.

$$D(S)_{ii} := \begin{cases} 1 & \text{if } i \in S \\ 0 & \text{otherwise.} \end{cases}$$

Note that $D(S^c) = I_n - D(S)$, since S^c is the complement of the set S. With this notation, we can represent P_S as

$$P_S = \{u \,:\, D(S)Xu \ge 0,\ (I_n - D(S))u \le 0\}\,,$$

and the maximization in the dual constraint as

$$\max_{u:\,\|u\|_2\le 1} v^T(Xu)_+ = \max_{\substack{S\subseteq\{1,\dots,n\}\\ S\in\mathcal{S}_X}} \max_{\substack{u:\,\|u\|_2\le 1\\ u\in P_S}} v^TD(S)Xu\,.$$

Enumerating all hyperplane arrangements $\mathcal{H}_X$, or equivalently $\mathcal{S}_X$, we index them in an arbitrary order via $i \in \{1,...,|\mathcal{S}_X|\}$. We denote $M = |\mathcal{S}_X|$. Hence, $S_1,...,S_M \in \mathcal{S}_X$ is the list of all M elements of $\mathcal{S}_X$. Next we use the strong duality result from Lemma 4 (see Appendix) for the inner maximization problem. The dual constraint (16) can be represented as

$$\begin{aligned} (16) \iff\ & \forall i \in \{1,...,M\} \text{ it holds that} \\ & \min_{\substack{\alpha\in\mathbb{R}^{|S_i|}\\ \beta\in\mathbb{R}^{|S_i^c|}\\ \alpha\ge 0\\ \beta\ge 0}} \|X^TD(S_i)\big(v+\alpha+\beta\big) - X^T\beta\|_2 \le \beta \\ \iff\ & \forall i \in \{1,...,M\}\ \exists(\alpha_i,\beta_i) \in \mathbb{R}^n \text{ s.t.} \\ & \alpha_i \ge 0\ \beta_i \ge 0 \\ & \|X^TD(S_i)\big(v+\alpha_i+\beta_i\big) - X^T\beta_i\|_2 \le \beta\,. \end{aligned}$$

Therefore, recalling the two-sided constraint in (15), we can represent the dual optimization problem in (15) as a finite dimensional convex optimization problem with variables $v \in \mathbb{R}^n$, $(\alpha_i, \beta_i, \alpha_i', \beta_i') \in \mathbb{R}^n$, $i = 1,...,M$, and $2M$

[2]See Appendix for generic convex loss functions

second order cone constraints as follows

$$\max_{\substack{v\in\mathbb{R}^n\\(\alpha_i,\beta_i)\in\mathbb{R}^n\\ \alpha_i,\beta_i\geq 0,\,\forall i\in[M]\\(\alpha_i',\beta_i')\in\mathbb{R}^n\\ \alpha_i',\beta_i'\geq 0,\,\forall i\in[M]}} \quad -\frac{1}{2}\|v-y\|_2^2+\frac{1}{2}\|y\|_2^2$$

$$\begin{aligned}\text{s.t. } & \|X^TD(S_1)\big(v+\alpha_1+\beta_1\big)-X^T\beta_1\|_2\leq\beta\\ & \vdots\\ & \|X^TD(S_M)\big(v+\alpha_M+\beta_M\big)-X^T\beta_M\|_2\leq\beta\\ & \|X^TD(S_1)\big(-v+\alpha_1'+\beta_1'\big)-X^T\beta_1'\|_2\leq\beta\\ & \vdots\\ & \|X^TD(S_M)\big(-v+\alpha_M'+\beta_M'\big)-X^T\beta_M'\|_2\leq\beta\end{aligned}$$

The above problem can be represented as a standard finite dimensional second order cone program. Note that the particular choice of parameters $v=0$, $\alpha_i=0$, $\beta_i=0$ for $i=1,...,M$ are strictly feasible in the above constraints as long as $\beta>0$. Therefore Slater's condition and consequently strong duality holds (Boyd & Vandenberghe, 2004a). The dual problem (15) can be written as

$$\min_{\substack{\lambda,\lambda'\in\mathbb{R}^M\\ \lambda,\lambda'\geq 0}}\ \max_{\substack{v\in\mathbb{R}^n\\(\alpha_i,\beta_i)\in\mathbb{R}^n\\ \alpha_i,\beta_i\geq 0,\,\forall i\in[M]\\(\alpha_i',\beta_i')\in\mathbb{R}^n\\ \alpha_i',\beta_i'\geq 0,\,\forall i\in[M]}} \quad -\frac{1}{2}\|v-y\|_2^2+\frac{1}{2}\|y\|_2^2$$
$$+\sum_{i=1}^M\lambda_i\big(\beta-\|X^TD(S_i)\big(v+\alpha_i+\beta_i\big)-X^T\beta_i\|_2\big)$$
$$+\sum_{i=1}^M\lambda_i'\big(\beta-\|X^TD(S_i)\big(-v+\alpha_i'+\beta_i'\big)-X^T\beta_i'\|_2\big).$$

Next we introduce variables $r_1,\dots,r_M,r_1',\dots,r_M'\in\mathbb{R}^d$ and represent the dual problem (15) as

$$\min_{\substack{\lambda,\lambda'\in\mathbb{R}^M\\ \lambda,\lambda'\geq 0}}\ \max_{\substack{v\in\mathbb{R}^n\\(\alpha_i,\beta_i)\in\mathbb{R}^n\\ \alpha_i,\beta_i\geq 0,\,\forall i\in[M]\\(\alpha_i',\beta_i')\in\mathbb{R}^n\\ \alpha_i',\beta_i'\geq 0,\,\forall i\in[M]}}\ \min_{\substack{r_i\in\mathbb{R}^d,\,\|r_i\|_2\leq 1\\ \forall i\in[M]}} -\frac{1}{2}\|v-y\|_2^2+\frac{1}{2}\|y\|_2^2$$
$$+\sum_{i=1}^M\lambda_i\big(\beta+r_i^TX^TD(S_i)\big(v+\alpha_i+\beta_i\big)-r_i^TX^T\beta_i\big)$$
$$+\sum_{i=1}^M\lambda_i'\big(\beta+{r_i'}^TX^TD(S_i)\big(-v+\alpha_i'+\beta_i'\big)-{r_i'}^TX^T\beta_i'\big).$$

We note that the objective is concave in v,α_i,β_i for $i=1,...,M$ and convex in $r_1,...r_M,r_1',\dots,r_M'$. Moreover the constraint sets $\|r_i\|_2\leq 1\,\|r_i'\|_2\leq 1\,\forall i$ are convex and compact. Invoking Sion's minimax theorem (Sion, 1958) for the inner $\max\min$ problem, we may express the strong dual of the problem (15) as

$$\min_{\substack{\lambda,\lambda'\in\mathbb{R}^M\\ \lambda,\lambda'\geq 0}}\ \min_{\substack{r_i\in\mathbb{R}^d,\,\|r_i\|_2\leq 1\\ r_i'\in\mathbb{R}^d,\,\|r_i'\|_2\leq 1\\ i=1,\dots,M}}\ \max_{\substack{v\in\mathbb{R}^n\\(\alpha_i,\beta_i)\in\mathbb{R}^n\\ \alpha_i,\beta_i\geq 0,\,\forall i\in[M]\\(\alpha_i',\beta_i')\in\mathbb{R}^n\\ \alpha_i',\beta_i'\geq 0,\,\forall i\in[M]}} -\frac{1}{2}\|v-y\|_2^2+\frac{1}{2}\|y\|_2^2$$
$$+\sum_{i=1}^M\lambda_i\big(\beta+r_i^TX^TD(S_i)\big(v+\alpha_i+\beta_i\big)-r_i^TX^T\beta_i\big)$$
$$+\sum_{i=1}^M\lambda_i'\big(\beta+{r_i'}^TX^TD(S_i)\big(-v+\alpha_i'+\beta_i'\big)-{r_i'}^TX^T\beta_i'\big).$$

Computing the maximum with respect to v,α_i,β_i for $i=1,...,M$ analytically we obtain the strong dual to (15) as

$$\min_{\substack{\lambda,\lambda'\in\mathbb{R}^M\\ \lambda,\lambda'\geq 0}}\ \min_{\substack{r_i\in\mathbb{R}^d,\,\|r_i\|_2\leq 1\\ r_i'\in\mathbb{R}^d,\,\|r_i'\|_2\leq 1\\ i=1,\dots,M\\(2D(S_i)-I_n)Xr_i\geq 0\\(2D(S_i)-I_n)Xr_i'\geq 0}} \frac{1}{2}\Big\|\sum_{i=1}^M\lambda_iD(S_i)Xr_i$$
$$-\lambda_i'D(S_i)Xr_i'-y\Big\|_2^2+\beta\sum_{i=1}^M(\lambda_i+\lambda_i')$$

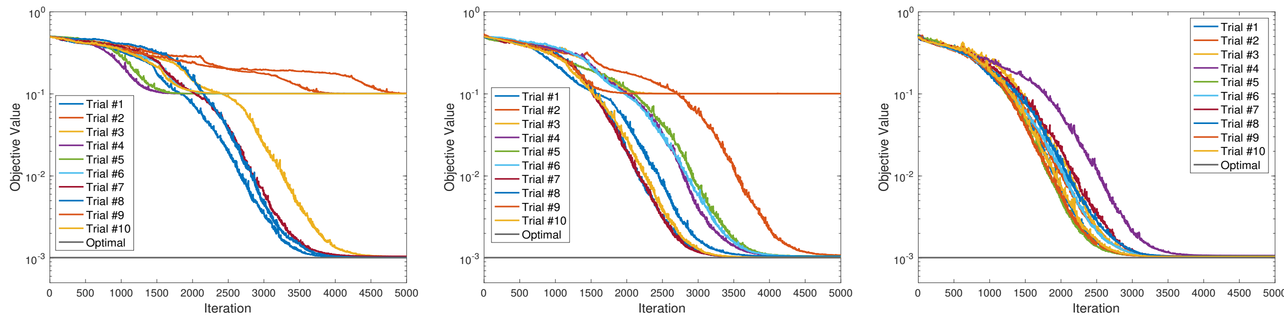
Now we apply a change of variables and define $w_i=\lambda_ir_i$ and $w_i'=\lambda_i'r_i'$ for $i=1,\dots,M$. Note that we can take $r_i=0$ when $\lambda_i=0$ without changing the optimal value. We obtain

$$\min_{\substack{w_i\in P_{S_i}\\ \|w_i\|_2\leq\lambda_i\\ w_i'\in P_{S_i}\\ \|w_i'\|_2\leq\lambda_i'\\ i=1,\dots,M\\ \lambda,\lambda'\geq 0}} \frac{1}{2}\Big\|\sum_{i=1}^MD(S_i)X(w_i-w_i')-y\Big\|_2^2+\beta\sum_{i=1}^M(\lambda_i+\lambda_i')$$

The variables λ_i, λ_i', $i=1,\dots,M$ can be eliminated since $\lambda_i=\|w_i\|_2$ and $\lambda_i'=\|w_i'\|_2$ are feasible and optimal. Plugging in these expressions, we get

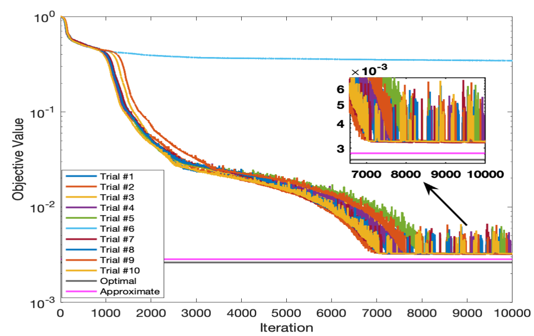
$$\min_{\substack{w_i\in P_{S_i}\\ w_i'\in P_{S_i}}} \frac{1}{2}\Big\|\sum_{i=1}^MD(S_i)X(w_i-w_i')-y\Big\|_2^2$$
$$+\beta\sum_{i=1}^M(\|w_i\|_2+\|w_i'\|_2)\,.$$

which is identical to (8), and proves that the objective values are equal. Given a solution to (8), we can form the network output as prescribed in the theorem statement, there will be m^* pairs $\{v_i^*,w_i^*\}$ for $i=1,\dots,P$, where either v_i^* or

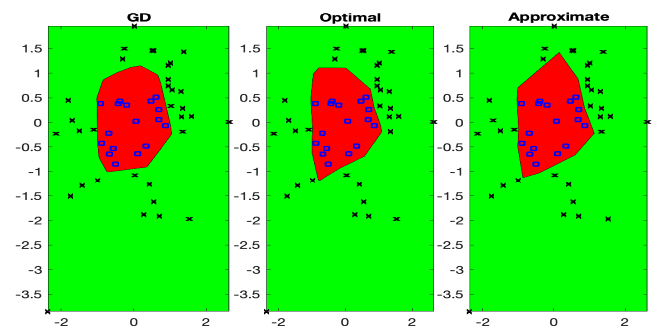


(a) $m = 8$ (b) $m = 15$ (c) $m = 50$

Figure 2: Training cost of a two-layer ReLU network trained with SGD (10 initialization trials) on a two dimensional dataset ($d = 2$), where Optimal denotes the objective value obtained by the proposed convex program in (8).



(a) Independent realizations with $m = 50$



(b) Decision boundaries

Figure 3: Training cost of a two-layer ReLU network trained with SGD (10 initialization trials) on a two-dimensional dataset, where Optimal and Approximate denote the objective value obtained by the proposed convex program in (8) and its approximation by sampling variables, respectively. Learned decision boundaries are also depicted.

w_i^* is non-zero since either the constraint $v^T(Xu^*) \leq \beta$, or $-v^T(Xu^*) \leq \beta$ can be active at any optima in (15). Constructing $\{u_i^*, \alpha_i^*\}_{i=1}^{m^*}$ as stated in the theorem, and plugging in the non-convex objective (2), we obtain the value

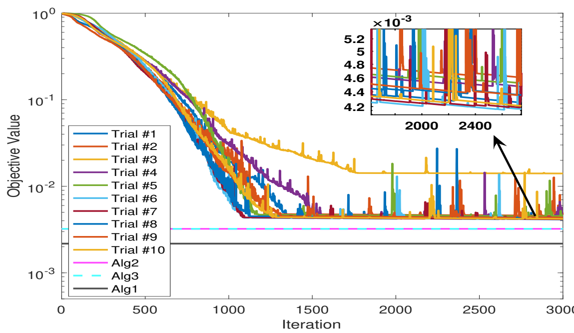
$$\begin{aligned} p^* \leq & \frac{1}{2}\Big\| \sum_{i=1}^{m^*} (Xu_i^*)_+ \alpha_i^* - y \Big\|_2^2 \\ & + \frac{\beta}{2} \sum_{i=1, v_i^* \neq 0}^{m^*} \Big\| \frac{v_i^*}{\sqrt{\|v_i^*\|_2}} \Big\|_2^2 + \Big\| \sqrt{\|v_i^*\|_2} \Big\|_2^2 \\ & + \frac{\beta}{2} \sum_{i=1, w_i^* \neq 0}^{m^*} \Big\| \frac{w_i^*}{\sqrt{\|w_i^*\|_2}} \Big\|_2^2 + \Big\| \sqrt{\|w_i^*\|_2} \Big\|_2^2 \end{aligned}$$

which is identical to the objective value of the convex program (8). Since the value convex program is equal to the value of it's dual d^* in (15), and $p^* \geq d^*$, we conclude that $p^* = d^*$, which is equal to the value of the convex program (8) achieved by the prescribed parameters. □

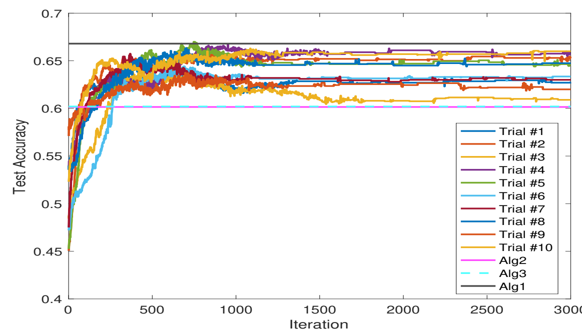
6. Numerical experiments

In this section, we present small scale numerical experiments to verify our results in the previous sections[3]. We first consider a one-dimensional dataset with $n = 5$, i.e., $X = [-2 \;\; -1 \; 0 \; 1 \; 2]^T$ and $y = [1 \;\; -1 \; 1 \; 1 \;\; -1]^T$. We then fit these data points using a two-layer ReLU network trained with SGD and the proposed convex program, where we use squared loss as a performance metric. In Figure 2, we plot the value of the regularized objective function with respect to the iteration index. Here, we plot 10 independent realizations for SGD and denote the convex program in (8) as "Optimal". Additionally, we repeat the same experiment for different number of neurons, particularly, $m = 8, 15$, and 50. As demonstrated in the figure, when the number of neurons is small, SGD is stuck at local minima. As we increase m, the number of trials that achieve the optimal performance gradually increases as well. We also note that Optimal achieves the smallest objective value as claimed in the previous sections. We then compare the performances

[3]Additional numerical results can be found in the Appendix

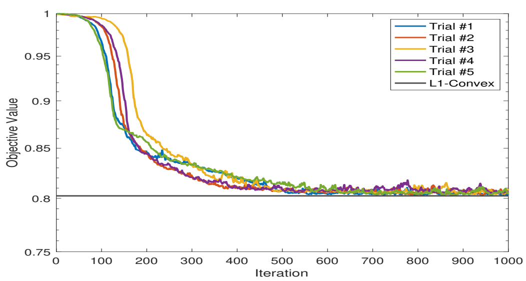


(a) Objective value

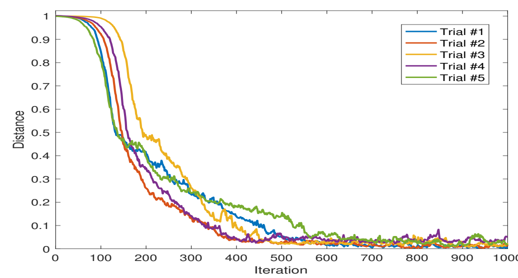


(b) Test accuracy

Figure 4: Training cost of a two-layer ReLU network trained with SGD (10 initialization trials) on a subset of CIFAR-10 and the proposed convex program (8) denoted as Alg1. Alg2 and Alg3 denote approximations of the proposed convex program by sampling variables.



(a) Objective value



(b) Distance to the solution of the convex program

Figure 5: Training accuracy of a two-layer linear CNN trained with SGD (5 initialization trials) on a subset of CIFAR-10, where L1-Convex denotes the proposed convex program in (14). Filters found via SGD converge to the solution of (14).

on two-dimensional datasets with $n = 50$, $m = 50$ and $y \in \{+1, -1\}^n$, where we use SGD with the batch size 25 and hinge loss as a performance metric. In these experiments, we also consider an approximate convex program, i.e., denoted as "Approximate" for which we use only a random subset of the diagonal matrices $D_1, ...D_P$ of size m. As illustrated in Figure 3, most of the SGD realizations converge to a slightly higher objective than Optimal. Interestingly, we also observe that even Approximate can outperform SGD in this case. In the same figure, we also provide the decision boundaries obtained by each method.

We also evaluate the performance of the algorithms on a small subset of CIFAR-10 for binary classification (Krizhevsky et al., 2014). Particularly, in each experiment, we first select two classes and then randomly under-sample to create a subset of the original dataset. For these experiments, we use hinge loss and SGD. In the first experiment, we train a two-layer ReLU network on the subset of CIFAR-10, where we include three different versions denoted as "Alg1", "Alg2", and "Alg3", respectively. For Alg1 and Alg2, we use a random subset of the diagonal matrices $D_1, ..., D_p$ which match the sign patterns of the initialized and optimized (by GD) network, respectively. For Alg3, we performed a heuristic adaptive sampling for the diagonal matrices: we first examine the values of Xu for each neuron using the initial weights and flip the sign pattern corresponding to small values and use it along with the original sign pattern. In Figure 4, we plot both the objective value and the corresponding test accuracy for 10 independent realizations with $n = 106$, $d = 100$, $m = 12$, and batch size 25. We observe that Alg1 achieves the lowest objective value and highest test accuracy. Finally, we train a two-layer linear CNN architecture on a subset of CIFAR-10, where we denote the proposed convex program in (14) as "L1-Convex". In Figure 5, we plot both the objective value and the Euclidean distance between the filters found by GD and L1-Convex for 5 independent realizations with $n = 387$, $m = 30$, $h = 10$, and batch size 60. In this experiment, all the realizations converge to the objective value obtained by L1-Convex and find almost the same filters.

7. Concluding remarks

We introduced a convex duality theory for non-convex neural network objectives and developed an exact representation via a convex program with polynomial many variables and constraints. Our results provide an equivalent characterization of neural network models in terms of convex regularization in a higher dimensional space where the data matrix is partitioned over all possible hyperplane arrangements. Neural networks, in fact behave precisely as convex regularizers, where piecewise linear models are fitted via an $\ell_1 - \ell_2$ group norm regularizer. There are a multitude of open research directions. One can obtain a better understanding of neural networks and their generalization properties by leveraging convexity, and high dimensional regularization theory (Wainwright, 2019). In the light of our results, one can view backpropagation as a heuristic method to solve the convex program (8), since the global minima are necessarily stationary points of the non-convex objective (2), i.e., fixed points of the update rule. Efficient optimization algorithms that approximate the convex program can be developed for larger scale experiments.

References

Agrawal, A., Verschueren, R., Diamond, S., and Boyd, S. A rewriting system for convex optimization problems. *Journal of Control and Decision*, 5(1):42–60, 2018.

Allen-Zhu, Z., Li, Y., and Song, Z. A convergence theory for deep learning via over-parameterization. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 242–252, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL `http://proceedings.mlr.press/v97/allen-zhu19a.html`.

Arora, R., Basu, A., Mianjy, P., and Mukherjee, A. Understanding deep neural networks with rectified linear units. In *6th International Conference on Learning Representations, ICLR 2018*, 2018.

Arora, S., Du, S. S., Hu, W., Li, Z., Salakhutdinov, R. R., and Wang, R. On exact computation with an infinitely wide neural net. In *Advances in Neural Information Processing Systems*, pp. 8139–8148, 2019.

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

Bengio, Y., Roux, N. L., Vincent, P., Delalleau, O., and Marcotte, P. Convex neural networks. In *Advances in neural information processing systems*, pp. 123–130, 2006.

Bienstock, D., Muñoz, G., and Pokutta, S. Principled deep neural network training through linear programming. *arXiv preprint arXiv:1810.03218*, 2018.

Boob, D., Dey, S. S., and Lan, G. Complexity of training relu neural network. *arXiv preprint arXiv:1809.10787*, 2018.

Boyd, S. and Vandenberghe, L. *Convex optimization*. Cambridge University Press, Cambridge, UK, 2004a.

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

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

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

Diamond, S. and Boyd, S. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

Du, S. S., Zhai, X., Poczos, B., and Singh, A. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=S1eK3i09YQ`.

Edelsbrunner, H., O'Rourke, J., and Seidel, R. Constructing arrangements of lines and hyperplanes with applications. *SIAM Journal on Computing*, 15(2):341–363, 1986.

Goberna, M. A. and López-Cerdá, M. *Linear semi-infinite optimization*. 01 1998. doi: 10.1007/978-1-4899-8044-1_3.

Goel, S., Kanade, V., Klivans, A., and Thaler, J. Reliably learning the relu in polynomial time. In Kale, S. and Shamir, O. (eds.), *Proceedings of the 2017 Conference on Learning Theory*, volume 65 of *Proceedings of Machine Learning Research*, pp. 1004–1042, Amsterdam, Netherlands, 07–10 Jul 2017. PMLR.

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

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pp. 8571–8580, 2018.

Krizhevsky, A., Nair, V., and Hinton, G. The CIFAR-10 dataset. `http://www.cs.toronto.edu/kriz/cifar.html`, 2014.

Neyshabur, B., Tomioka, R., and Srebro, N. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*, 2014.

Ojha, P. C. Enumeration of linear threshold functions from the lattice of hyperplane intersections. *IEEE Transactions on Neural Networks*, 11(4):839–850, 2000.

Recht, B., Fazel, M., and Parrilo, P. A. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review*, 52(3):471–501, 2010.

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

Rosset, S., Swirszcz, G., Srebro, N., and Zhu, J. L1 regularization in infinite dimensional feature spaces. In *International Conference on Computational Learning Theory*, pp. 544–558. Springer, 2007.

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

Savarese, P., Evron, I., Soudry, D., and Srebro, N. How do infinite width bounded norm networks look in function space? *CoRR*, abs/1902.05040, 2019. URL http://arxiv.org/abs/1902.05040.

Shapiro, A. Semi-infinite programming, duality, discretization and optimality conditions. *Optimization*, 58(2):133–161, 2009.

Sion, M. On general minimax theorems. *Pacific J. Math.*, 8(1):171–176, 1958. URL https://projecteuclid.org:443/euclid.pjm/1103040253.

Stanley, R. P. et al. An introduction to hyperplane arrangements. *Geometric combinatorics*, 13:389–496, 2004.

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

Tütüncü, R., Toh, K., and Todd, M. 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.

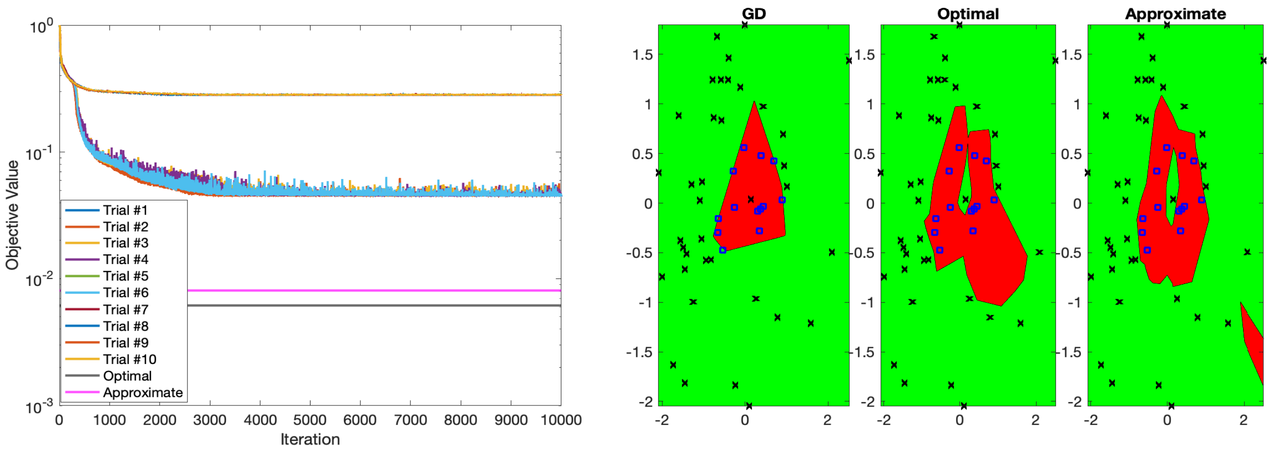
Wainwright, M. J. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

Winder, R. Partitions of n-space by hyperplanes. *SIAM Journal on Applied Mathematics*, 14(4):811–818, 1966.

Yuan, M. and Lin, Y. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68 (1):49–67, 2006.

A. Appendix

A.1. Additional numerical results



(a) Independent SGD initialization trials with $m = 50$

(b) Decision boundaries

Figure 6: Training of a two-layer ReLU network with SGD (10 initialization trials) and proposed convex programs on a two-dimensional dataset. Optimal and Approximate denote the objective value obtained by the proposed convex program (8) and its approximation, respectively. Learned decision boundaries are also depicted.

We now present another numerical experiment on a two-dimensional dataset[4], where we place a negative sample ($y = -1$) near the positive samples ($y = +1$) to have a more challenging loss landscape. In Figure 6, we observe that all the SGD realizations are stuck at local minima, therefore, achieve a significantly higher objective value compared to both Optimal and Approximate, which are based on convex optimization.

In addition to the classification datasets, we evaluate the performance of the algorithms on three regression datasets, i.e., the Boston Housing, Kinematics, and Bank datasets (Torgo). In Figure 7, we plot the objective value and the corresponding test error of 5 independent initialization trials with respect to time in seconds, where we use squared loss and choose $n = 400$, $d = 13$, $m = 12$, and batch size(bs) 25. Similarly, we plot the objective values and test errors for the Kinematics and Bank datasets in Figure 8 and 9, where $(n, d, m, \text{bs}) = (4000, 8, 12, 25)$ and $(n, d, m, \text{bs}) = (4000, 32, 12, 25)$, respectively. We observe that Alg1 achieves the lowest objective value and test error in both cases.

We also consider the training of the following two-layer CNN architecture

$$\min_{\{\alpha_j, u_j\}_{j=1}^m} \frac{1}{2}\Big\| \sum_{k=1}^{K} \sum_{j=1}^{m} (X_k u_j)_+ \alpha_{jk} - y_k \Big\|_2^2 + \frac{\beta}{2} \sum_{j=1}^{m} (\|u_j\|_2^2 + \|\alpha_j\|_2^2)\,,$$

which has an equivalent convex program as follows

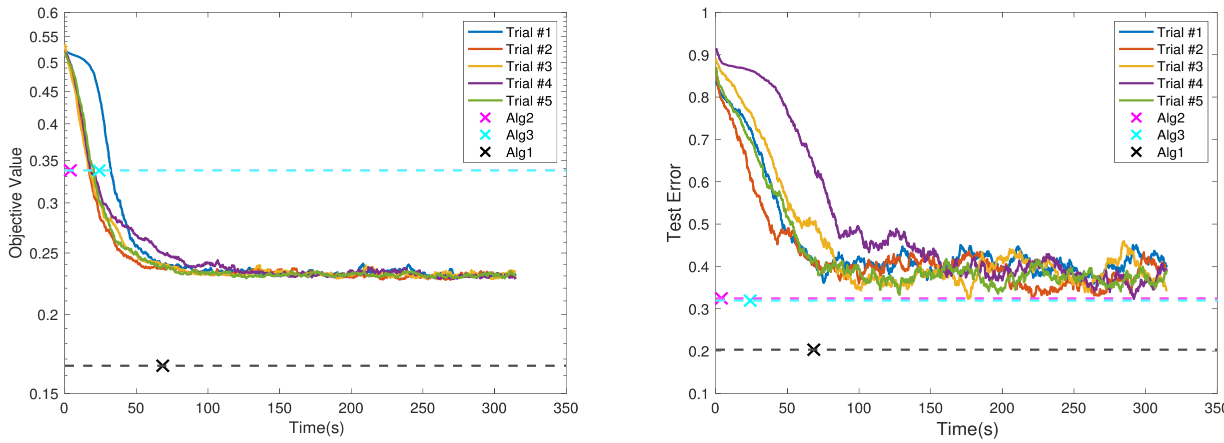
$$\min_{\{w_j, w_j'\}_{j=1}^m} \frac{1}{2}\Big\| \sum_{k=1}^{K} \sum_{j=1}^{m} D(S_{jk}) X_k (w_j - w_j') - y_k \Big\|_2^2 + \beta \sum_{j=1}^{m} (\|w_j\|_2 + \|w_j'\|_2)$$
$$\text{s.t. } (2D(S_{jk}) - I_n) X_k w_j \geq 0,\ (2D(S_{jk}) - I_n) X_k w_j' \geq 0,\ \|w_j\|_2 \leq 1,\ \|w_j'\|_2 \leq 1,\ \forall j \in [m], \forall k \in [K].$$

In Figure 10, we provide the binary classification performance of the algorithms on a subset of CIFAR-10, where we use hinge loss and choose $(n, d, m, \text{bs}) = (195, 3072, 50, 20)$, filter size $4 \times 4 \times 3$, and stride 4. This experiment also illustrates that Alg1 achieves lower objective value and higher test accuracy compared with the other methods including GD. We also emphasize that in this experiment, we use sign patterns of a clustered subset of patches, specifically 50 clusters, as well as the GD patterns for Alg1. As depicted in Figure 11, the neurons that correspond to the sign patterns of GD matches with the

[4] In all the experiments, we use CVX (Grant & Boyd, 2014) and CVXPY (Diamond & Boyd, 2016; Agrawal et al., 2018) with the SDPT3 solver (Tütüncü et al., 2001) to solve convex optimization problems.

neurons found by GD. Thus, the performance difference stems from the additional sign patterns found by clustering the patches.

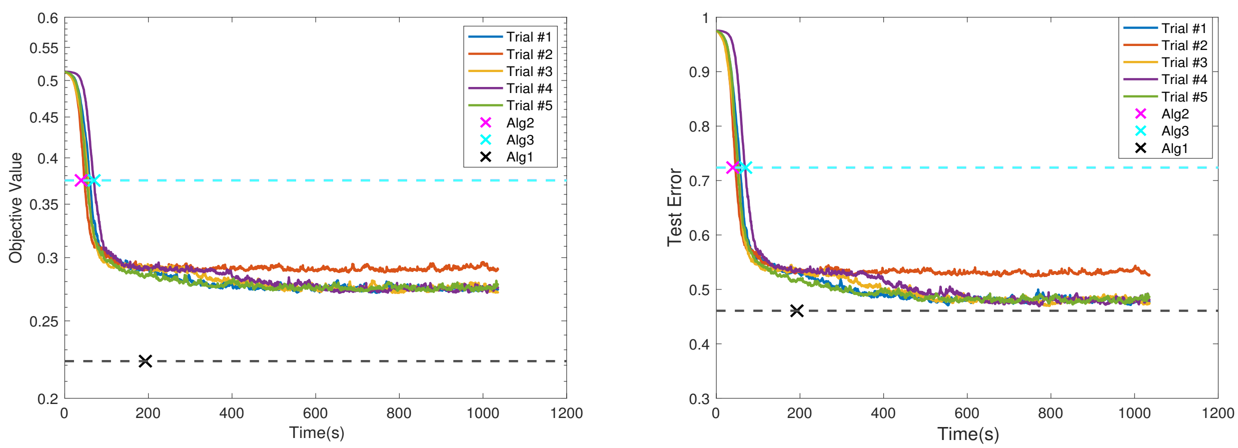
In order to evaluate the computational complexity of the introduced approaches, in Table 1, we provide the training time of each algorithm in the main paper. This data clearly shows that the introduced convex programs outperform GD while requiring significantly less training time.



(a) Training objective value

(b) Test error

Figure 7: Training and test errors of the algorithms on the Boston Housing dataset ($n = 400$ and $d = 13$) where we run SGD independently in 5 initialization trials. For the convex program (8) (Alg1), and its approximations (Alg2 and Alg3), crossed markers correspond to the total computation time of the convex optimization solver.



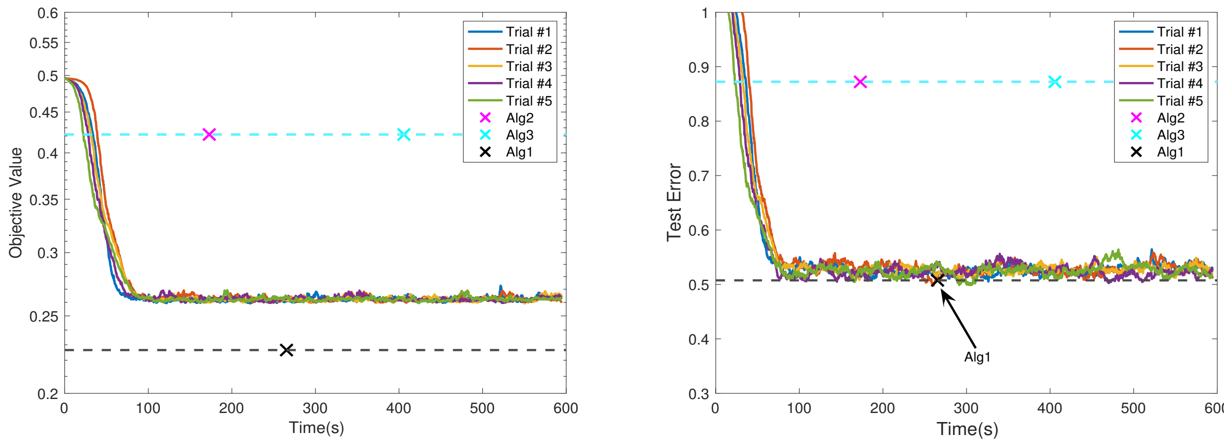
(a) Training objective value

(b) Test error

Figure 8: Performance comparison of the algorithms on the Kinematics dataset ($n = 4000$ and $d = 8$) where we run SGD independently in 5 initialization trials. For the convex program (8) (Alg1), and its approximations (Alg2 and Alg3), crossed markers correspond to the total computation time of the convex optimization solver.

A.2. Constructing hyperplane arrangements in polynomial time

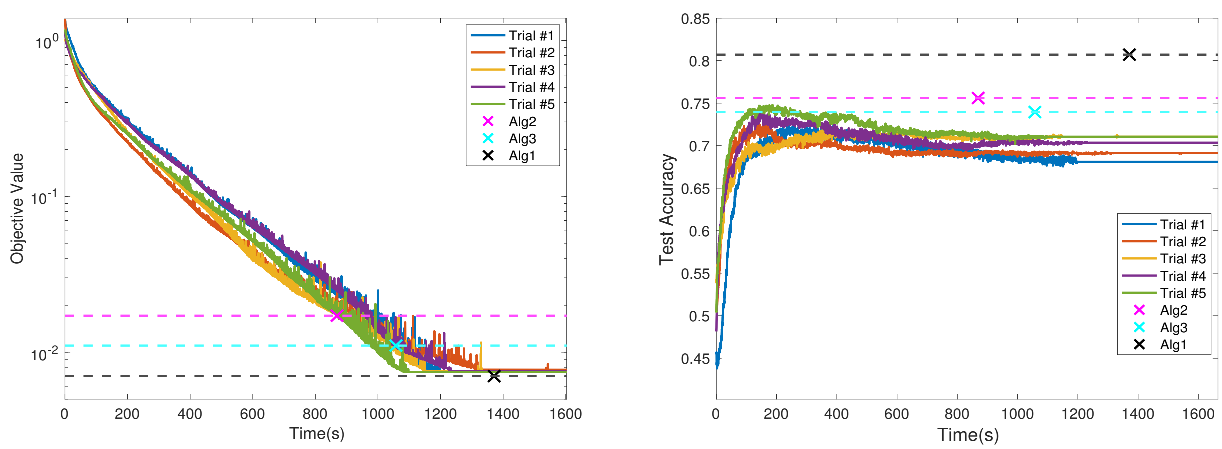
We now consider the number of all distinct sign patterns $\text{sign}(Xz)$ for all possible choices $z \in \mathbb{R}^d$. Note that this number is the number of regions in a partition of $\mathbb{R}^d$ by hyperplanes passing through the origin, and are perpendicular to the rows of X. We now show that the dimension d can be replaced with $\text{rank}(X)$ without loss of generality. Suppose that the data matrix X has rank r. We may express $X = U\Sigma V^T$ using its Singular Value Decomposition in compact form, where $U \in \mathbb{R}^{n\times r}, \Sigma \in \mathbb{R}^{r\times r}, V^T \in \mathbb{R}^{r\times d}$. For any vector $z \in \mathbb{R}^d$ we have $Xz = U\Sigma V^T z = Uz'$ for some $z' \in \mathbb{R}^r$. Therefore,



(a) Training objective value

(b) Test error

Figure 9: Performance comparison of the algorithms on the Bank dataset ($n = 4000$ and $d = 32$) where we run SGD independently in 5 initialization trials. For the convex program (8) (Alg1), and its approximations (Alg2 and Alg3), crossed markers correspond to the total computation time of the convex optimization solver.



(a) Training objective value

(b) Test accuracy

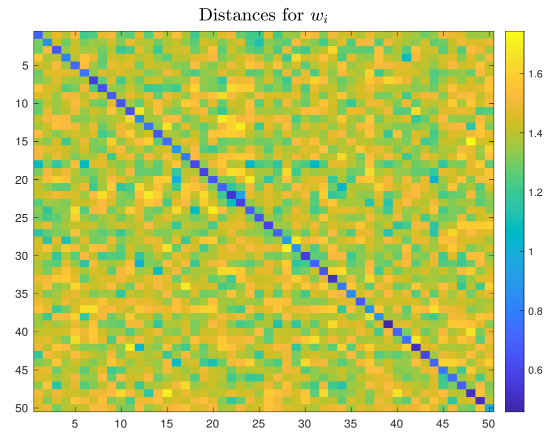
Figure 10: Performance of the algorithms for two-layer CNN training on a subset of CIFAR-10 ($n = 195$ and filter size $4 \times 4 \times 3$) where we run SGD independently in 5 initialization trials. For the convex program (8) (Alg1), and its approximations (Alg2 and Alg3), crossed markers correspond to the total computation time of the convex optimization solver.

the number of distinct sign patterns $\text{sign}(Xz)$ for all possible $z \in \mathbb{R}^d$ is equal to the number of distinct sign patterns $\text{sign}(Uz')$ for all possible $z \in \mathbb{R}^r$.

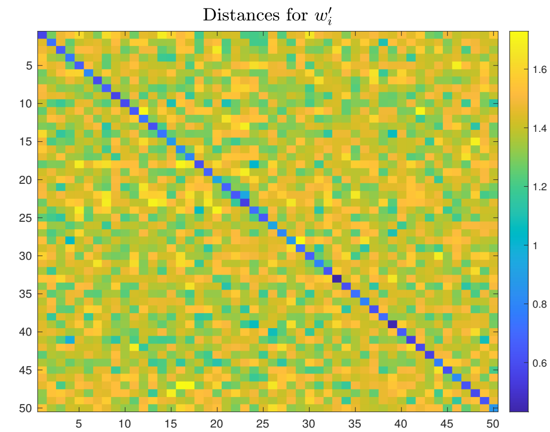
Consider an arrangement of n hyperplanes $\in \mathbb{R}^r$, where $n \geq r$. Let us denote the number of regions in this arrangement by $P_{n,r}$. In (Ojha, 2000; Cover, 1965) it's shown that this number satisfies

$$P_{n,r} \leq 2 \sum_{k=0}^{r-1} \binom{n-1}{k}. \tag{17}$$

For hyperplanes in general position, the above inequality is in fact an equality. In (Edelsbrunner et al., 1986), the authors present an algorithm that enumerates all possible hyperplane arrangements $O(n^r)$ time, which can be used to construct the data for the convex program (8).



(a)



(b)

Figure 11: Visualization of the distance (using the Euclidean norm of the difference) between the neurons found by GD and our convex program in Figure 10. The ij^{th} entries of the distance plots are $\left\|\frac{w_i}{\|w_i\|_2} - \frac{u_j}{\|u_j\|_2}\right\|_2$ and $\left\|\frac{w_i'}{\|w_i'\|_2} - \frac{u_j}{\|u_j\|_2}\right\|_2$, respectively.

Table 1: Training time(in seconds), final objective value and test accuracy($\%$) of each algorithm in the main paper,where we use the CVX SDPT3 solver to optimize the convex programs.

	Figure 2		Figure 3			Figure 4				Figure 5	
	SGD	Optimal	GD	Approx.	Optimal	GD	Alg1	Alg2	Alg3	GD	L1-Convex
Time(s)	420.663	1.225	890.339	1.498	117.858	624.787	108.065	5.931	12.009	65.365	1.404
Train. Objective	0.001	0.001	0.0032	0.0028	0.0026	0.0042	0.0022	0.0032	0.0032	0.804	0.803
Test Accuracy($\%$)	-	-	-	-	-	62.75	66.80	60.15	60.20	-	-

A.3. Equivalence of the ℓ_1 penalized neural network training cost

In this section, we prove the equivalence between (2) and (3).

Lemma 2 ((Savarese et al., 2019; Neyshabur et al., 2014))**.** The following two problems are equivalent:

$$\min_{\{u_j,\alpha_j\}_{j=1}^m} \frac{1}{2}\left\|\sum_{j=1}^m (Xu_j)_+\alpha_j - y\right\|_2^2 + \frac{\beta}{2}\sum_{j=1}^m (\|u_j\|_2^2 + \alpha_j^2) = \min_{\|u_j\|_2\le 1} \min_{\{\alpha_j\}_{j=1}^m} \frac{1}{2}\left\|\sum_{j=1}^m (Xu_j)_+\alpha_j - y\right\|_2^2 + \beta\sum_{j=1}^m |\alpha_j|\,,.$$

[**Proof of Lemma 2**] We can rescale the parameters as $\bar{u}_j = \gamma_j u_j$ and $\bar{\alpha}_j = \alpha_j/\gamma_j$, for any $\gamma_j > 0$. Then, the output becomes

$$\sum_{j=1}^m (X\bar{u}_j)_+\bar{\alpha}_j = \sum_{j=1}^m (Xu_j\gamma_j)_+\frac{\alpha_j}{\gamma_j} = \sum_{j=1}^m (Xu_j)_+\alpha_j,$$

which proves that the scaling does not change the network output. In addition to this, we have the following basic inequality

$$\frac{1}{2}\sum_{j=1}^m (\alpha_j^2 + \|u_j\|_2^2) \ge \sum_{j=1}^m (|\alpha_j|\;\|u_j\|_2),$$

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

Now, let us consider a modified version of the problem, where the unit norm equality constraint is relaxed as $\|u_j\|_2 \leq 1$. Let us also assume that for a certain index j, we obtain $\|u_j\|_2 < 1$ with $\alpha_j \neq 0$ as an optimal solution. This shows that the unit norm inequality constraint is not active for u_j, and hence removing the constraint for u_j will not change the optimal solution. However, when we remove the constraint, $\|u_j\|_2 \to \infty$ reduces the objective value since it yields $\alpha_j = 0$. Therefore, we have a contradiction, which proves that all the constraints that correspond to a nonzero α_j must be active for an optimal solution. This also shows that replacing $\|u_j\|_2 = 1$ with $\|u_j\|_2 \leq 1$ does not change the solution to the problem.

A.4. Dual problem for (3)

The following lemma proves the dual form of (3).

Lemma 3. The dual form of the following primal problem

$$\min_{\|u_j\|_2 \leq 1} \min_{\{\alpha_j\}_{j=1}^m} \frac{1}{2}\Big\| \sum_{j=1}^m (Xu_j)_+ \alpha_j - y \Big\|_2^2 + \beta \sum_{j=1}^m |\alpha_j| \,,$$

is given by the following

$$\min_{\|u_j\|_2 \leq 1} \max_{\substack{v \in \mathbb{R}^n \text{ s.t.} \\ |v^T (Xu_j)_+| \leq \beta}} -\frac{1}{2}\|y - v\|_2^2 + \frac{1}{2}\|y\|_2^2 \,.$$

[**Proof of Lemma 3**] Let us first reparametrize the primal problem as follows

$$\min_{\|u_j\|_2 \leq 1} \min_{r, \{\alpha_j\}_{j=1}^m} \frac{1}{2}\|r\|_2^2 + \beta \sum_{j=1}^m |\alpha_j| \text{ s.t. } r = \sum_{j=1}^m (Xu_j)_+ \alpha_j - y,$$

which has the following Lagrangian

$$L(v, r, u_j, \alpha_j) = \frac{1}{2}\|r\|_2^2 + \beta \sum_{j=1}^m |\alpha_j| + v^T r + v^T y - v^T \sum_{j=1}^m (Xu_j)_+ \alpha_j .$$

Then, minimizing over r and α yields the proposed dual form.

A.5. Dual problem for (11)

Let us first reparameterize the primal problem as follows

$$\max_{M, v} -\frac{1}{2}\|v - y\|_2^2 + \frac{1}{2}\|y\|_2^2 \text{ s.t. } \sigma_{\max}(M) \leq \beta,\ M = [X_1^T v \dots X_K^T v].$$

Then the Lagrangian is as follows

$$\begin{aligned} L(\lambda, Z, M, v) &= -\frac{1}{2}\|v - y\|_2^2 + \frac{1}{2}\|y\|_2^2 + \lambda(\beta - \sigma_{\max}(M)) + \operatorname{trace}(Z^T M) - \operatorname{trace}(Z^T [X_1^T v \dots X_K^T v]) \\ &= -\frac{1}{2}\|v - y\|_2^2 + \frac{1}{2}\|y\|_2^2 + \lambda(\beta - \sigma_{\max}(M)) + \operatorname{trace}(Z^T M) - v^T \sum_{k=1}^K X_k z_k \end{aligned}$$

where $\lambda \geq 0$. Then maximizing over M and v yields the following dual form

$$\min_{z_k \in \mathbb{R}^d, \forall k \in [K]} \frac{1}{2}\Big\| \sum_{k=1}^K X_k z_k - y \Big\|_2^2 + \beta \Big\| [z_1, \dots, z_K] \Big\|_*,$$

where $\Big\|[z_1, \dots, z_K]\Big\|_* = \|Z\|_* = \sum_i \sigma_i(Z)$ is the ℓ_1 norm of singular values, i.e., nuclear norm (Recht et al., 2010).

A.6. Dual problem for (13)

Let us denote the eigenvalue decomposition of U_j as $U_j = FD_jF^H$, where $F \in \mathbb{C}^{d\times d}$ is the Discrete Fourier Transform matrix and $D_j \in \mathbb{C}^{d\times d}$ is a diagonal matrix. Then, applying the scaling in Lemma 2 and then taking the dual as in Lemma 3 yields

$$\max_v -\frac{1}{2}\|v-y\|_2^2 + \frac{1}{2}\|y\|_2^2 \text{ s.t. } \|v^T XFDF^H\|_2 \leq \beta,\ \forall D : \|D\|_F^2 \leq 1$$

which can be equivalently written as

$$\max_v -\frac{1}{2}\|v-y\|_2^2 + \frac{1}{2}\|y\|_2^2 \text{ s.t. } \|v^T \tilde{X} D\|_2 \leq \beta,\ \forall D : \|D\|_F^2 \leq 1.$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Since D is diagonal, $\|D\|_F^2 \leq 1$ is equivalent to $\sum_{i=1}^d D_{ii}^2 \leq 1$. Therefore, the problem above can be further simplified as

$$\max_v -\frac{1}{2}\|v-y\|_2^2 + \frac{1}{2}\|y\|_2^2 \text{ s.t. } \|v^T \tilde{X}\|_\infty \leq \beta\ .$$

Then, taking the dual of this problem gives the following

$$\min_{z\in\mathbb{C}^d} \frac{1}{2}\left\|\tilde{X}z - y\right\|_2^2 + \beta\|z\|_1.$$

A.7. Dual problem for vector output two-layer linear convolutional networks

Vector version of the two-layer linear convolutional network training problem has the following dual

$$\max_V \ \operatorname{trace} V^T Y \tag{18}$$

$$\text{s.t. } \max_{\|u\|_2\leq 1} \sum_k \|V^T X_k u\|_2^2 \leq 1 \tag{19}$$

Similarly, extreme points are the maximal eigenvectors of $\sum_k X_k^T V V^T X_k$ When $V = Y$, and one-hot encoding is used, these are the right singular vectors of the matrix $[X_{1,c}^T\, X_{2,c}^T \ldots X_{K,c}^T]^T$ whose rows contain all the patch vectors for class c.

A.8. Semi-infinite strong duality

Note that the semi-infinite problem (4) is convex. We first show that the optimal value is finite. For $\beta > 0$, it is clear that $v = 0$ is strictly feasible, and achieves 0 objective value. Since $-\|y - v\|_2^2 \leq 0$, the optimal objective value p^* satisfies $p^* \leq \|y\|_2^2$, and hence $0 \leq d^* \leq \frac{1}{2}\|y\|_2^2$. Consequently, Theorem 2.2 of (Shapiro, 2009) implies that strong duality holds, i.e., $p^* = d^*_\infty$, if the solution set of the semi-infinite problem in (4) is nonempty and bounded. Next, we note that the solution set of (4) is the Euclidean projection of y onto the polar set $(\mathcal{Q}_X \cup -\mathcal{Q}_X)^\circ$ which is a convex, closed and bounded set since the function $\big(Xu\big)_+$ can be expressed as the union of finitely many convex closed and bounded sets. $\square$

A.9. Semi-infinite strong gauge duality

Now we prove strong duality for (7). We invoke the semi-infinite optimality conditions for the dual (7), in particular we apply Theorem 7.2 of (Goberna & López-Cerdá, 1998) and use the standard notation therein. We first define the set

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

Note that $\mathbf{K}$ is the union of finitely many convex closed sets, since the function $\big(Xu\big)_+$ can be expressed as the union of finitely many convex closed sets. Therefore the set $\mathbf{K}$ is closed. By Theorem 5.3 (Goberna & López-Cerdá, 1998), this implies that the set of constraints in (15) forms a Farkas-Minkowski system. By Theorem 8.4 of (Goberna & 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 (15). $\square$

A.10. Neural Gauge function and equivalence to minimum norm networks

Consider the gauge function

$$p^g = \min_{r \geq 0} r \tag{20}$$
$$\text{s.t. } ry \in \text{conv}(\mathcal{Q}_X \cup -\mathcal{Q}_X). \tag{21}$$

and its dual representation in terms of the support function of the polar of $\text{conv}(\mathcal{Q}_X \cup -\mathcal{Q}_X)$.

$$d^g = \max_v v^T y \tag{22}$$
$$v \in (\mathcal{Q}_X \cup -\mathcal{Q}_X)^\circ. \tag{23}$$

Since the set $\mathcal{Q}_X \cup -\mathcal{Q}_X$ is a closed convex set that contains the origin, we have $p^g = d^g$ (Rockafellar, 1970) and $\big(\text{conv}(\mathcal{Q}_X \cup -\mathcal{Q}_X)\big)^\circ = (\mathcal{Q}_X \cup -\mathcal{Q}_X)^\circ$. The result in Section A.8 implies that the above value is equal to the semi-infinite dual value, i.e., $p^d = p^g_\infty$, where

$$p^g_\infty := \min_\mu \|\mu\|_{TV} \text{ s.t. } \int_{u \in \mathcal{B}_2} (Xu)_+ d\mu(u) = y\,. \tag{24}$$

By Caratheodory's theorem, there exists optimal solutions the above problem consisting of m^* Dirac deltas (Rockafellar, 1970; Rosset et al., 2007), and therefore

$$p^g_\infty = \min_{u_j \in \mathcal{B}_2, j \in [m^*]} \sum_{j=1}^{m^*} |\alpha_j| \text{ s.t. } \sum_{j=1}^{m^*} (Xu_j)_+ d\alpha_j = y\,, \tag{25}$$

where we define m^* as the number of Dirac delta's in the optimal solution to p^g_∞. If the optimizer is non-unique, we define m^* as the minimum cardinality solution among the set of optimal solutions. Now consider the non-convex problem

$$\min_{\{u_j, \alpha_j\}_{j=1}^m} \|\alpha\|_1 \tag{26}$$
$$\text{s.t. } \sum_{j=1}^m (Xu_j)_+ \alpha_j = y \tag{27}$$
$$\|u_j\|_2 \leq 1\,. \tag{28}$$

Using the standard parameterization for ℓ_1 norm we get

$$\min_{\{u_j\}_{j=1}^m, s \geq 0, t \geq 0} \sum_{j=1}^m (t_j + s_j) \tag{29}$$
$$\text{s.t. } \sum_{j=1}^m (Xu_j)_+ t_j - (Xu_j)_+ s_j = y \tag{30}$$
$$\|u_j\|_2 \leq 1\,, \forall j. \tag{31}$$

Introducing a slack variable $r \in \mathbb{R}_+$, an equivalent representation can be written as

$$\min_{\{u_j\}_{j=1}^m, s \geq 0, t \geq 0, r \geq 0} r \tag{32}$$
$$\text{s.t. } \sum_{j=1}^m (Xu_j)_+ t_j - (Xu_j)_+ s_j = y \tag{33}$$
$$\sum_{j=1}^m (t_j + s_j) = r \tag{34}$$
$$\|u_j\|_2 \leq 1\,, \forall j. \tag{35}$$

Note that $r > 0$ as long as $y \neq 0$. Rescaling variables by letting $t'_j = t_j/r$, $s'_j = s_j/r$ in the above program, we obtain

$$\begin{aligned} &\min_{\{u_j\}_{j=1}^m, s'\geq 0, t'\geq 0, r\geq 0} \quad r \\ &\text{s.t.} \ \sum_{j=1}^m \big((Xu_j)_+ t'_j - (Xu_j)_+ s'_j\big) = ry \\ &\sum_{j=1}^m (t'_j + s'_j) = 1 \\ &\|u_j\|_2 \leq 1\,, \forall j\,. \end{aligned} \tag{36}$$

Suppose that $m \geq m^*$. It holds that

$$\exists s', t' \geq 0\,, \{u_j\}_{j=1}^m \ \text{ s.t. } \sum_{j=1}^m (t'_j + s'_j) = 1,\ \|u_j\|_2 \leq 1,\ \forall j \sum_{j=1}^m (Xu_j)t'_j - (Xu_j)_+ s'_j = ry \quad \iff \quad ry \in \text{conv}(\mathcal{Q}_X \cup -\mathcal{Q}_X). \tag{37}$$

We conclude that the optimal value of (36) is identical to the gauge function p_g.

A.11. Alternative proof of the semi-infinite strong duality

It holds that $p^* \geq d^*$ by weak duality in (4). Theorem 1 proves that the objective value of (15) is identical to the value of (2) as long as $m \geq m^*$. Therefore we have $p^* = d^*$. □

A.12. Finite Dimensional Strong Duality Results for Theorem 1

Lemma 4. Suppose $D(S)$, $D(S^c)$ are fixed diagonal matrices as described earlier, and X is a fixed matrices. The dual of the convex optimization problem

$$\max_{\substack{u\in\mathbb{R}^d \\ \|u\|_2\leq 1 \\ D(S)Xu\geq 0 \\ D(S^c)Xu\leq 0}} \quad v^T D(S) X u$$

is given by

$$\min_{\substack{\alpha\in\mathbb{R}^{|S|} \\ \beta\in\mathbb{R}^{|S^c|} \\ \alpha\geq 0 \\ \beta\geq 0}} \ \|X^T D(S)\big(v + \alpha + \beta\big) - X^T\beta\|_2\,,$$

and strong duality holds.

Note that the linear inequality constraints specify valid hyperplane arrangements. Then there exists strictly feasible points in the constraints of the maximization problem. Standard finite second order cone programming duality implies that strong duality holds (Boyd & Vandenberghe, 2004b) and the dual is as specified. □

A.13. General loss functions

In this section, we extend our derivations to arbitrary convex loss functions.

Consider minimizing the sum of the squared loss objective and squared ℓ_2-norm of all parameters

$$p^* := \min_{\{\alpha_j, u_j\}_{j=1}^m} \ell\left(\sum_{j=1}^m (Xu_j)_+ \alpha_j, y\right) + \frac{\beta}{2}\sum_{j=1}^m (\|u_j\|_2^2 + \alpha_j^2)\,. \tag{38}$$

where where $\ell(\cdot, y)$ is a convex loss function. Then, consider the following finite dimensional convex optimization problem

$$\begin{aligned}
&\min_{\{v_i,w_i\}_{i=1}^P} \ \ell\left(\sum_{i=1}^P D_i X(v_i - w_i), y\right) + \beta \sum_{i=1}^P \left(\|v_i\|_2 + \|w_i\|_2\right) \\
&\text{s.t.} \\
&(2D_i - I)Xv_i \geq 0,\ (2D_i - I)Xw_i \geq 0, \forall i \in [P], \qquad (39)
\end{aligned}$$

Let us define $m^* := \sum_{j: v_j^* \neq 0 \text{ or } u_j^* \neq 0}^P 1$, where $\{v_i^*, w_i^*\}_{i=1}^P$ are optimal in (39).

Theorem 5. The convex program (39) and the non-convex problem (38) where $m \geq m^*$ has identical optimal values. Moreover, an optimal solution to (38) can be constructed from an optimal solution to (39) as follows

$$u_i^* = \begin{cases} \frac{v_i^*}{\sqrt{\|v_i^*\|_2}},\ \alpha_j^* = \sqrt{\|v_i^*\|_2} & \text{if} \quad \|v_i^*\|_2 > 0 \\ \frac{w_i^*}{\sqrt{\|w_i^*\|_2}},\ \alpha_j^* = -\sqrt{\|w_i^*\|_2} & \text{otherwise}\,, \end{cases}$$

where v_i^*, w_i^* are the optimal solutions to (39), and either v_i^* or w_i^* is non-zero for all $i = 1, ..., P$.

[**Proof of Theorem 5**] The proof parallels the proof of the main result section and Theorem 6. We note that dual constraint set remains the same, and analogous strong duality results apply as we show next.

We also show that our dual characterization holds for arbitrary convex loss functions.

$$\min_{\{u_j,\alpha_j\}_{j=1}^m} \ \ell\left(\sum_{j=1}^m (Xu_j)_+ \alpha_j, y\right) + \beta\|\alpha\|_1 \text{ s.t. } \|u_j\|_2 \leq 1,\ \forall j, \qquad (40)$$

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

Theorem 6. The dual of (40) is given by

$$\max_v -\ell^*(v) \text{ s.t. } |v^T (Xu)_+| \leq \beta,\ \forall u \in \mathcal{B}_2\,,$$

where ℓ^* is the Fenchel conjugate function defined as

$$\ell^*(v) = \max_z z^T v - \ell(z, y)\,.$$

[**Proof of Theorem 6**] The proof follows from classical Fenchel duality (Boyd & Vandenberghe, 2004b). We first describe (40) in an equivalent form as follows

$$\min_{z,\{u_j,\alpha_j\}_{j=1}^m} \ell(z, y) + \beta\|\alpha\|_1 \text{ s.t. } z = \sum_{j=1}^m (Xu_j)_+\alpha_j,\ \|u_j\|_2 \leq 1, \forall j.$$

Then the dual function is

$$g(v) = \min_{z,\{u_j,\alpha_j\}_{j=1}^m} \ell(z, y) - v^T z + v^T \sum_{j=1}^m (Xu_j)_+\alpha_j + \beta\|\alpha\|_1 \text{ s.t. } \|u_j\|_2 \leq 1, \forall j.$$

Therefore, using the classical Fenchel duality (Boyd & Vandenberghe, 2004b) yields the claimed dual form.