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 training two-layer neural networks with rectified linear units (ReLUs) in terms of a single convex program with number of variables polynomial in the number of training samples and the number of hidden neurons. Our theory utilizes semi-infinite duality and minimum norm regularization. We show that ReLU networks trained with standard weight decay are equivalent to block ℓ_1 penalized convex models. Moreover, we show that certain standard convolutional linear networks are equivalent semi-definite programs which can be simplified to ℓ_1 regularized linear models in a polynomial sized discrete Fourier feature space.

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 (ReLU) 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.

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 modeling capabilities of neural networks for any finite number of hidden neurons. We prove that networks with ReLU are identical to *convex regularization* methods in a finite higher dimensional space.

Consider a two-layer network $f : \mathbb{R}^d \to \mathbb{R}$ with m neurons

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

where $u_j \in \mathbb{R}^d$ and $\alpha_j \in \mathbb{R}$ are the weights for hidden and output layers, respectively, and $\phi(t) = (t)_+ := \max(t, 0)$ is the ReLU activation. We extend the definition of scalar functions to vectors/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 $[n]$. We also 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 other network architectures (see Appendix).

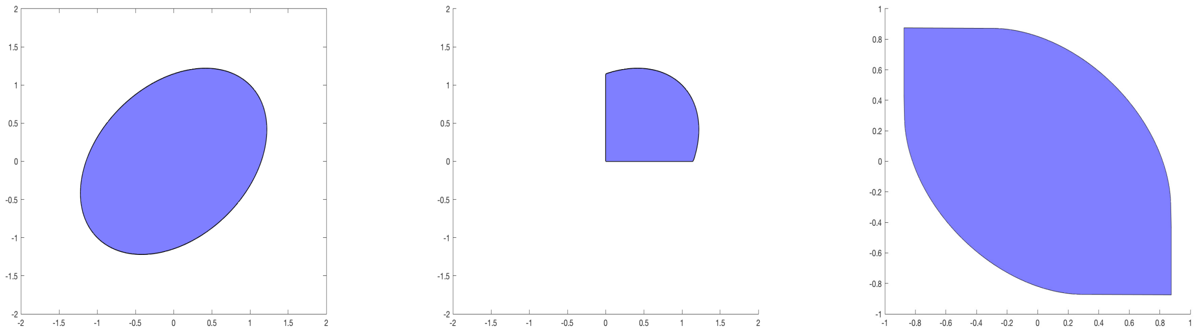
Given a data matrix $X \in \mathbb{R}^{n\times d}$, a label vector $y \in \mathbb{R}^n$, and a regularization parameter $\beta > 0$, consider minimizing the squared loss objective and squared ℓ_2-norm of all parameters

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

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

[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, Online, PMLR 119, 2020. Copyright 2020 by the author(s).



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

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

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

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

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 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 A.3),

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

Replacing the inner minimization problem with its convex dual, we obtain (see Appendix A.4)

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

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

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

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 $m \ge m^*$ for some $m^* \in \mathbb{N}$, $1 \le m^* \le n$, 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 are 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 $|q(u)| \le 1$. 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 measure. And the total variation norm $\|\mu\|_{TV}$ of μ reduces to the ℓ_1 norm $\|\alpha\|_1$.

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

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

Furthermore, an application of Caratheodory's theorem shows that the infinite dimensional bi-dual (5) always has a solution that is supported with m^* Dirac deltas, 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

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

We show 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 A.10), i.e.,

$$p^*_{\beta\to 0} = \inf_{t:t\geq 0} \; t \text{ s.t. } y \in t\,\mathrm{Conv}\{\mathcal{Q}_X \cup -\mathcal{Q}_X\} \,,$$

which we call *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

$$p^*_{\beta\to 0} = \max y^T z \text{ s.t. } z \in (\mathcal{Q}_X \cup -\mathcal{Q}_X)^\circ \,, \qquad (7)$$

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 A.2).

Consider the finite dimensional convex problem

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

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

$$\begin{aligned} (u^*_{j_{1i}}, \alpha^*_{j_{1i}}) &= \left(\frac{v_i^*}{\sqrt{\|v_i^*\|_2}}, \sqrt{\|v_i^*\|_2} \right) \qquad \text{if} \quad v_i^* \neq 0 \\ (u^*_{j_{2i}}, \alpha^*_{j_{2i}}) &= \left(\frac{w_i^*}{\sqrt{\|w_i^*\|_2}}, -\sqrt{\|w_i^*\|_2} \right) \quad \text{if} \quad w_i^* \neq 0 \,, \end{aligned}$$

where $\{v_i^*, w_i^*\}_{i=1}^P$ are the optimal solutions to (8). Thus, we have $m^* = \sum_{i:v_i^* \neq 0}^P 1 + \sum_{i:w_i^* \neq 0}^P 1$.

Remark 3.1. *Note that optimal solutions of* (8) *may not be unique. As an example, merging positively colinear neurons does not change the objective value. Particularly, if there exist positively colinear solutions such as $v_{i_1}^* = c_1 v_i$ and $v_{i_2}^* = c_2 v_i$, where $v_i \in \mathbb{R}^d$ and $c_1, c_2 \in \mathbb{R}_+$, then merging these solutions as $v_i^* = (c_1 + c_2) v_i$ yields the same objective.*

Remark 3.2. *Theorem 1 shows that two-layer ReLU networks with m hidden neurons can be globally optimized via the second order cone program* (8) *with $2dP$ variables and $2nP$ linear inequalities where $P = 2r\Big(\frac{e(n-1)}{r}\Big)^r$, and r = rank(X). The computational complexity is at most $O\Big(d^3 r^3 \big(\frac{n}{r}\big)^{3r}\Big)$ 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*

[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.

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). It is interesting to note that the proposed convex program trains ReLU neural networks* optimally*, unlike local search heuristics such as backpropagation, which may converge to suboptimal solutions (see Section 6 for numerical evidence). To the best of our knowledge, our results provide the first polynomial-time algorithm to train non-trivial neural networks with global optimality guarantees. 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. Theorem 1 implies that ReLU networks are equivalent to sparse mixtures of linear models, where sparsity is enforced by the group* $\ell_1 - \ell_2$ *convex regularizer. More specifically, the non-convex neural network approach implicitly maps the data to the higher dimensional feature matrix* $[D_1X, ..., D_PX]$*, and consequently seeks a group sparse model.*

Remark 3.3. *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* $D_i = Diag(1[Xu_i \geq 0])$*,* $\forall i \in [\tilde{P}]$ *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 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 backpropagation, 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 can be described by patch matrices $X_k \in \mathbb{R}^{n\times d}$, $k = 1, ..., K$. This formulation also includes image, or tensor inputs. For flattened activations, we have $f(X_1, ..., X_K) = \sum_{j=1}^m \sum_{k=1}^K \phi(X_k u_j)\alpha_{jk}$ as the network output. We first present a simpler case for vector regression, $f_k(X_1, ..., X_K) = \sum_{j=1}^m \phi(X_k u_j)\alpha_j$ which is separable over the patch index k.

4.1. Separable ReLU convolutional networks

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_k's are labels. We first note that this problem is separable over the patch indices k, therefore, do not exactly correspond to classical convolutional network architectures which are not separable. The above problem can be reduced to the fully connected case (2) by defining $X' = [X_1^T, ..., X_K^T]^T$ and $y' = [y_1^T, ..., y_K^T]^T$. Therefore, the convex program (8) solves the above problem exactly in $O\Big(d^3 r^3 \big(\frac{n}{r}\big)^{3r}\Big)$ complexity, where r is the number of variables in a single filter. Note that typical CNNs use m filters of size 3 × 3 (r=9) in the first layer (He et al., 2016).

4.2. Linear convolutional network training as a Semi-definite Program (SDP)

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

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

The corresponding dual problem is given by

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

Similar arguments to those used in the proof of Theorem 1, strong duality holds. 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

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

The dual of the above SDP is a nuclear norm penalized convex optimization problem (see Appendix A.5)

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

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

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

where $U_j \in \mathbb{R}^{d\times d}$ is a circulant matrix generated by a circular shift modulo d using the elements $u_j \in \mathbb{R}^h$. Then, the SDP (11) reduces to (see Appendix A.6)

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

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

5. Proof of the Main Result (Theorem 1)

We now prove the main result 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

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

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

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

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

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

Enumerating all hyperplane arrangements $\mathcal{H}_X$, or equivalently $\mathcal{S}_X$, we index them in an arbitrary order via $i \in [|\mathcal{S}_X|]$. 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 for the inner maximization problem. The dual constraint (16) can be represented as

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

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 $v \in \mathbb{R}^n, \alpha_i, \beta_i, \alpha_i', \beta_i' \in \mathbb{R}^n, \forall i \in [M]$, and $2M$ second

[2] See Appendix A.13 for generic convex loss functions.

order cone constraints as follows

$$\begin{aligned}\max_{\substack{v\in\mathbb{R}^n\\ \alpha_i,\beta_i\in\mathbb{R}^n\\ \alpha_i,\beta_i\geq 0,\,\forall i\in[M]\\ \alpha_i',\beta_i'\in\mathbb{R}^n\\ \alpha_i',\beta_i'\geq 0,\,\forall i\in[M]}} \quad & -\frac{1}{2}\|v-y\|_2^2+\frac{1}{2}\|y\|_2^2\\ \text{s.t. } & \|X^TD(S_1)\big(v+\alpha_1+\beta_1\big)-X^T\beta_1\|_2\leq\beta\\ & \vdots\\ & \|X^TD(S_M)\big(v+\alpha_M+\beta_M\big)-X^T\beta_M\|_2\leq\beta\\ & \|X^TD(S_1)\big(-v+\alpha_1'+\beta_1'\big)-X^T\beta_1'\|_2\leq\beta\\ & \vdots\\ & \|X^TD(S_M)\big(-v+\alpha_M'+\beta_M'\big)-X^T\beta_M'\|_2\leq\beta.\end{aligned}$$

The above problem can be represented as a standard finite dimensional second order cone program. Note that the particular choice of parameters $v=\alpha_i=\beta_i=\alpha_i'=\beta_i'=0$, $\forall i\in[M]$, 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

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

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

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

We note that the objective is concave in v,α_i,β_i and convex in r_i,r_i', $\forall i\in[M]$. 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

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

Computing the maximum with respect to $v,\alpha_i,\beta_i,\alpha_i',\beta_i'$, $\forall i\in[M]$, analytically we obtain the strong dual to (15) as

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

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

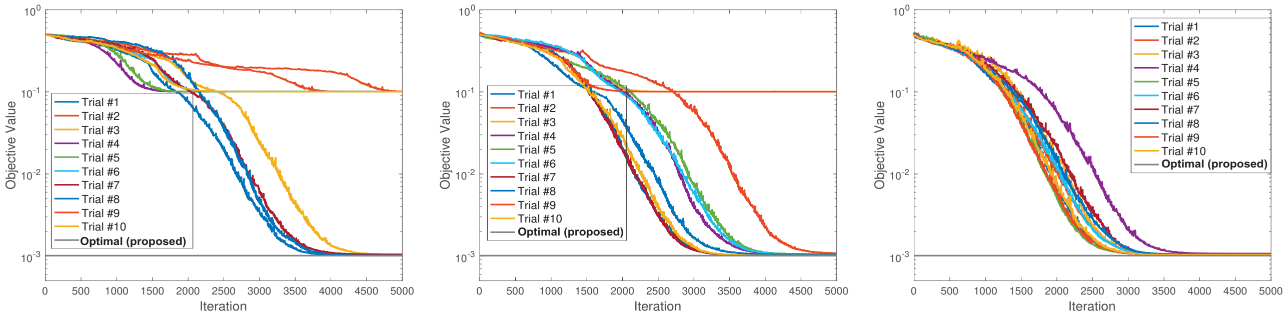
$$\min_{\substack{w_i,w_i'\in P_{S_i}\\ \|w_i\|_2\leq\lambda_i\\ \|w_i'\|_2\leq\lambda_i'\\ \lambda,\lambda'\geq 0}}\frac{1}{2}\Big\|\sum_{i=1}^MD(S_i)X(w_i'-w_i)-y\Big\|_2^2+\beta\sum_{i=1}^M(\lambda_i+\lambda_i').$$

The variables λ_i, λ_i', $\forall i\in[M]$ 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

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

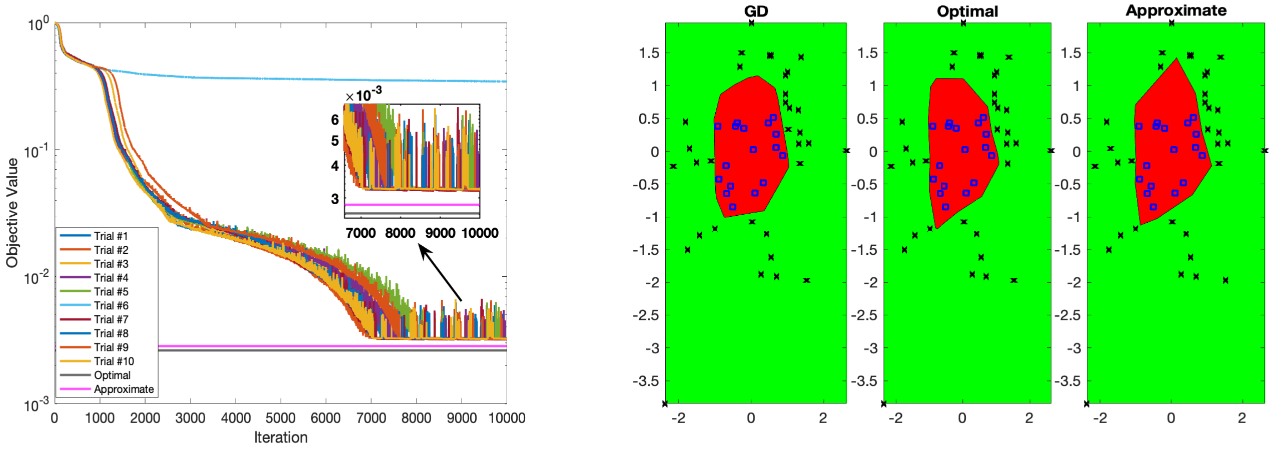
which is identical to (8), and proves that the objective values are equal. Constructing $\{u_j^*,\alpha_j^*\}_{j=1}^{m^*}$ as stated in the theorem, and plugging in (2), we obtain the value

$$\begin{aligned}p^*\leq&\frac{1}{2}\Big\|\sum_{j=1}^{m^*}(Xu_j^*)_+\alpha_j^*-y\Big\|_2^2\\ &+\frac{\beta}{2}\sum_{i=1,v_i^*\neq 0}^{P}\left(\Big\|\frac{v_i^*}{\sqrt{\|v_i^*\|_2}}\Big\|_2^2+\Big\|\sqrt{\|v_i^*\|_2}\Big\|_2^2\right)\end{aligned}$$



(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 one dimensional dataset ($d = 1$), where Optimal denotes proposed convex programming approach in (8). SGD can be stuck at local minima for small m, while the proposed approach is optimal as guaranteed by Theorem 1.



(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.

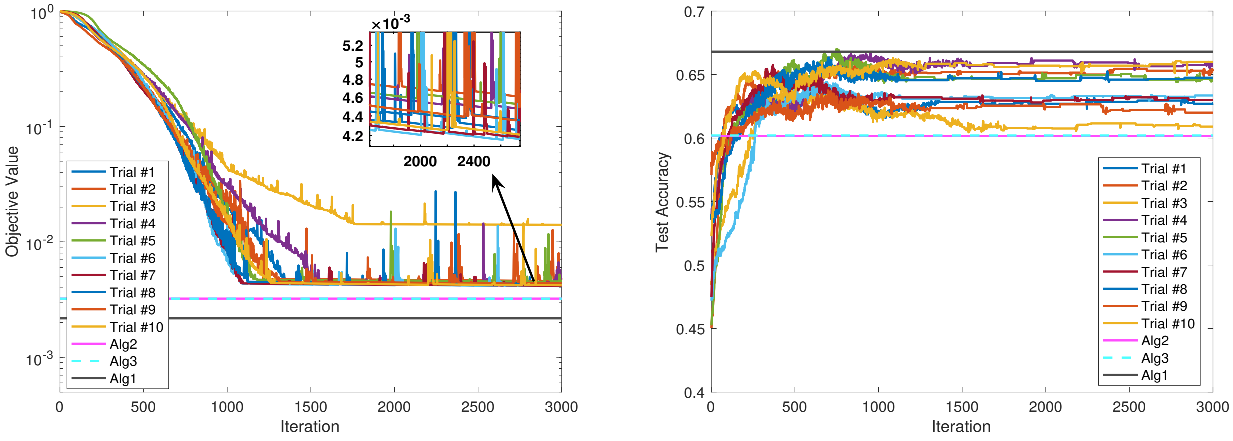
$$+\frac{\beta}{2}\sum_{i=1,w_i^*\neq 0}^{P}\left(\left\|\frac{w_i^*}{\sqrt{\|w_i^*\|_2}}\right\|_2^2+\left\|\sqrt{\|w_i^*\|_2}\right\|_2^2\right),$$

which is identical to the objective value of the convex program (8). Since the value of the 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 $y = [1 \; -1 \; 1 \; 1 \; -1]^T$, where we include the bias term by simply concatenating a column of ones to the data matrix X. 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, which is also consistent with the interpretations in (Ergen & Pilanci, 2019). We also note that Optimal achieves the smallest objective value as claimed in the previous sections. We then compare the performances

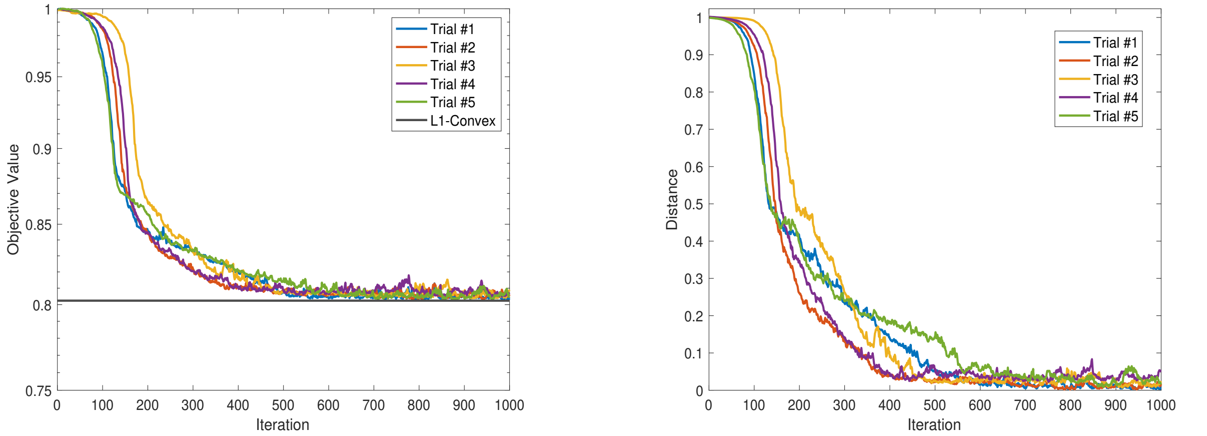
[3] Additional experiments can be found in the Appendix A.1



(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 convex program (8) denoted as Alg1. Alg2 and Alg3, which are approximations of the convex program.



(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 $D_1, \ldots D_P$ 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, we use a random subset of the diagonal matrices $D_1, ..., D_p$ which match the sign patterns of the optimized (by SGD) network along with a randomly selected subset of possible sign patterns. Similarly, for Alg2, we use the sign patterns that match the initialized network. For Alg3, we perform 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. ReLU neural networks can be precisely represented as convex regularizers, where piecewise linear models are fitted via an $\ell_1 - \ell_2$ group norm regularizer. It is well known that two-layer networks have a quite rich representation power thanks to their universal approximation property. However, our results clearly show that the fitted model is parsimonious due to the $\ell_1 - \ell_2$ group regularization, which facilitates better generalization. Thus, we believe that our characterization sheds light into the extraordinary success of ReLU networks. 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) and analyze the loss landscape, since the global minima are necessarily stationary points of the non-convex objective (2), i.e., fixed points of the update rule. Interesting consequences in this direction are reported in (Lacotte & Pilanci, 2020) after our work. Furthermore, one can extend our convex approach to various architectures, e.g., modern CNNs, recurrent networks, and autoencoders. Based on our methods, recently (Ergen & Pilanci, 2020a) considered convex programs for CNNs with various pooling strategies, and determined several other convex regularizers implied by the network architecture. Finally, to the best of our knowledge, our results provide the first algorithm to train non-trivial neural networks *optimally*. On the other hand, the popular backpropagation method is a local search heuristic, which may not find the optimal neural network and may be dramatically inefficient as shown in the experiments. Efficient optimization algorithms that exactly or approximately solve the convex program can be developed for larger scale experiments, including proximal and stochastic gradient methods.

Acknowledgements

This work was supported in part by the National Science Foundation under grant IIS-1838179 and Stanford SystemX Alliance.

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.

Ergen, T. and Pilanci, M. Convex optimization for shallow neural networks. In *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 79–83, 2019.

Ergen, T. and Pilanci, M. Training convolutional relu neural networks in polynomial time: Exact convex optimization formulations. *arXiv preprint arXiv:2006.14798*, 2020a.

Ergen, T. and Pilanci, M. Convex geometry of two-layer relu networks: Implicit autoencoding and interpretable models. In Chiappa, S. and Calandra, R. (eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 4024–4033, Online, 26–28 Aug 2020b. PMLR. URL `http://proceedings.mlr.press/v108/ergen20a.html`.

Ergen, T. and Pilanci, M. Convex geometry and duality of over-parameterized neural networks. *arXiv preprint arXiv:2002.11219*, 2020c.

Ergen, T. and Pilanci, M. Convex duality of deep neural networks. *arXiv preprint arXiv:2002.09773*, 2020d.

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.

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

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.