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 convolutional linear networks are equivalent to \( \ell_1 \) 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 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 generalization and modeling capabilities of neural networks 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

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

where \( u_1, \ldots, u_m \in \mathbb{R}^d \) and \( \alpha_1, \ldots, \alpha_m \in \mathbb{R} \) are the weights for the hidden and output layers, respectively, and \( \phi(t) = (t)_+ := \max(t, 0) \) is the ReLU activation function. We extend the definition of scalar functions to vectors and matrices entry-wise. We use \( B_p \) to denote the unit \( \ell_p \) ball in \( \mathbb{R}^d \). We denote the set of indices from 1 to \( n \) as \([n]\). Furthermore, we use \( \sigma \) 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 \( X \in \mathbb{R}^{n \times d} \), label vector \( y \in \mathbb{R}^n \), and a regularization parameter \( \beta > 0 \), consider minimizing the squared loss objective and squared \( \ell_2 \)-norm of all parameters

\[
p^* := \min_{\{\alpha_j, u_j\}_{j=1}^m} \frac{1}{2} \left( \sum_{j=1}^m (X u_j)_+ + \alpha_j - y \right)^2 + \frac{\beta}{2} \sum_{j=1}^m (\|u_j\|^2 + \alpha_j^2).
\]

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

Neural Networks are Convex Regularizers

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 $O(2^m n^m)$ (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 (3). We start with the $\ell_1$ penalized representation, which is equivalent to (2) (see Appendix A.4).

$$p^* = \min_{\|w\|_1 \leq 1} \min_{v, j \in [m]} \max_{\alpha_j \in \mathbb{N}} \frac{1}{2} \|\sum_{j=1}^{m} (Xu_j) + \alpha_j - y\|_2^2 + \beta \sum_{j=1}^{m} |\alpha_j|,$$

(3)

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

$$p^* = \min_{\|w\|_1 \leq 1} \max_{v, j \in [m]} \max_{x \in \mathbb{R}^d \setminus \{0\}} \frac{1}{2} \|y - v\|_2^2 + \frac{1}{2} \|y\|_2^2.$$  

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

$$p^* \geq d^* := \max_{v \in \mathbb{R}^d \setminus \{0\}, \|v\|_2 \leq \beta} \min_{\mu} \frac{1}{2} \|y - v\|_2^2 + \frac{1}{2} \|y\|_2^2,$$

(4)

where $B_2$ is the unit $\ell_2$ 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 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 $\phi_{u_j} : \mathcal{X} \rightarrow \mathbb{R}$ parameterized by $u \in B_2$. We then consider real-valued Radon measures equipped with the uniform norm ($\|\cdot\|_V$) (Rudin, 1964). For a signed Radon measure $\mu$, we can define an infinite width neural network output for the input $x \in \mathcal{X}$ as $f(x) = \int_{u \in B_2} \phi_u(x) d\mu(u)$. The total variation norm of the signed measure $\mu$ is defined as the supremum of $\int_{u \in B_2} |\phi_u(x)| d\mu(u)$ over all continuous functions $\phi(u)$ that satisfy $|\phi(u)| \leq 1$. Consider the ReLU basis functions $\phi_u(x) = (x^T u)_+$. 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 $\delta$ is the Dirac delta measure. And the total variation norm $\|\mu\|_{TV}$ of $\mu$ reduces to the $\ell_1$ 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

$$d^* \leq p^* \leq \min_{\mu} \frac{1}{2} \left\| \int_{u \in B_2} (Xu)_+ d\mu(u) - y \right\|_2^2 + \beta \|\mu\|_{TV}.$$  

(5)
where \( \|\mu\|_{TV} \) stands for the total variation norm of the Radon measure \( \mu \). Furthermore, an application of Caratheodory’s theorem shows that the infinite dimensional bi-dual \(^5\) always has a solution that is supported with finite Dirac deltas, whose exact number we define as \( m^* \), where \( m^* \leq n + 1 \) \cite{Rosset2007}. Therefore we have

\[
\begin{aligned}
p_{\infty}^* &= \min_{\|u\|_2 \leq 1} \frac{1}{2} \left( \sum_{j=1}^{m^*} (X u_j)^+ \alpha_j - y \right) \left\| u \right\|_2^2 + \beta \sum_{j=1}^{m^*} |\alpha_j|,

\end{aligned}
\]

as long as \( m \geq m^* \). We show that strong duality holds, i.e., \( d^* = p^* \) in Appendix \[A.10\] 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} (X u_j)^+ \alpha_j = y, \|u_j\|_2 \leq 1 \forall j.

\end{aligned}
\]

It can be shown that \( p_{\beta \to 0}^* \) is the gauge function of the convex hull of \( Q_X \cup -Q_X \), where \( Q_X := \{(X u)_+ : u \in B_2\} \) (see Appendix \[A.10\]), i.e.,

\[
p_{\beta \to 0}^* = \inf_{t \geq 0} t \text{ s.t. } y \in t \text{ Conv}\{Q_X \cup -Q_X\},
\]

which we call Neural Gauge due to the connection to the minimum norm interpolation problem. Using classical polar gauge duality \cite{Rockafellar1970}, it holds that

\[
\begin{aligned}
p_{\beta \to 0}^* = \max_{y \in X^*} y^T z \text{ s.t. } z \in (Q_X \cup -Q_X)^o,

\end{aligned}
\]

where \((Q_X \cup -Q_X)^o\) is the polar of the set \( Q_X \cup -Q_X \). 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 \((Q_X \cup -Q_X)^o\) is always convex \cite{Rockafellar1970}, 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 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 \( \text{Diag}(1|X u \geq 0|) \) where \( u \in \mathbb{R}^d \) is arbitrary and \( 1|X u \geq 0| \in \{0,1\}^n \) is an indicator vector with Boolean elements \([1|x_i^Tu \geq 0|, \ldots, 1|x_n^Tu \geq 0|]\). Let us enumerate all such distinct diagonal matrices that can be obtained for all possible \( u \in \mathbb{R}^d \), and denote them as \( D_1, \ldots, D_p \). \( P \) 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

\[
\begin{aligned}
P \leq 2 \sum_{k=0}^{r-1} \binom{n-1}{k} \leq 2r \left( e(n-1) \right)^r,

\end{aligned}
\]

for \( r \leq n \) where \( r := \text{rank}(X) \). \cite{Ohba2000, Stanley2004, Winder1966, Cover1965} \( \text{see Appendix \[A.2\]} \). Consider the finite dimensional convex problem

\[
\begin{aligned}
\min_{\{v_i,w_i\}_{i=1}^p} & \frac{1}{2} \sum_{i=1}^{p} D_i X (v_i - w_i) - y \left\| v_i \right\|_2^2 \\
\text{s.t.} & (2D_i - I_n)Xv_i \geq 0, (2D_i - I_n)Xw_i \geq 0, \forall i.

\end{aligned}
\]

**Theorem 1.** The convex program \((8)\) and the non-convex problem \((2)\) with \( m^* \) neurons can be constructed from an optimal solution to \((8)\) as follows

\[
\begin{aligned}
(u^*_i, \alpha^*_i) = \begin{cases} \\
\left( -\frac{v^*_i}{\sqrt{|v^*_i|}}, \sqrt{|v^*_i|} \right) & \text{if } v^*_i \neq 0 \\
\left( \frac{w^*_i}{\sqrt{|w^*_i|}}, -\sqrt{|w^*_i|} \right) & \text{if } w^*_i \neq 0,
\end{cases}
\end{aligned}
\]

where \( v^*_i, w^*_i \) are the optimal solutions to \((8)\), and either \( v^*_i \) or \( w^*_i \) is non-zero, \( \forall i \in [P] \). We have \( m^* = \sum_{j=0}^{P} v^*_j \neq 0 \lor w^*_j \neq 0 \), where \( \{v^*_i, w^*_i\}_{i=1}^P \) are optimal in \((9)\).

**Remark 3.1.** Theorem \[7\] shows that two-layer ReLU networks with \( h \) hidden neurons can be globally optimized via the second order cone program \((3)\) with \( 2dP \) variables and \( 2nP \) linear inequalities where \( P = 2r \left( e(n-1) \right)^r \), and \( r = \text{rank}(X) \). The computational complexity is \( O(d^3r^3(\frac{n}{r})^{3r}) \) 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 \cite{Ahrg2018, Bienstock2018}. Note that \( d \) is a small number that corresponds to the filter size in CNNs as we illustrate in the next section. However, \(^4m^* \) is defined as the number of Dirac deltas in the optimal solution to \((4)\). If the optimum is not unique, we may pick the minimum cardinality solution.
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 (Huan & Lin 2009) 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 $D_1, ..., D_p$. For example, one can generate $u \sim N(0, I_d)$, or from any distribution $P$ times, and let $D_k = \text{Diag}(1|Xu_i \geq 0|), \forall i \in [P]$ 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 (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 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, $f_k(X_1, ..., X_K) = \sum_{j=1}^m \phi(X_k u_j)\alpha_{jk}$ which is separable over the $k$ index.

4.1. ReLU convolutional networks with vector outputs

Consider the training problem

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

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

where $y_k$’s are 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 $O(d^3 r^3 + \frac{n}{r} )$ complexity, where $r$ is the number of variables in a single filter. Note that typical CNNs use $m$ filters of size $3 \times 3$ ($r=9$) in the first hidden layer (He et al. 2016).

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

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

$$\min_{\{u_j, \alpha_j\}_{j=1}^m} \frac{1}{2} \left\| \sum_{k=1}^K \sum_{j=1}^m X_k u_j \alpha_{jk} - y \right\|_2^2$$

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

The corresponding dual problem is given by

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

(10)

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

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

(11)

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

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

(12)

where $\left\| [z_1, ..., z_K] \right\|_* = \|Z\|_* := \sum_i \sigma_i(Z)$ is the $\ell_1$ 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

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

(13)

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

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

(14)

where $\tilde{X} = X F$, 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. We start with the dual representation

\[
\begin{align*}
\max & -\frac{1}{2} \|v - y\|_2^2 + \frac{1}{2} \|y\|_2^2 \\
\text{s.t.} & \max_{u: \|u\|_2 \leq 1} \left| v^T (X_u)^+ \right| \leq \beta.
\end{align*}
\tag{15}
\]

Note that the constraint (15) can be represented as

\[
\left\{ v : \max_{\|u\|_2 \leq 1} v^T (X_u)^+ \leq \beta \right\} \cap \left\{ v : \max_{\|u\|_2 \leq 1} -v^T (X_u)^+ \leq \beta \right\}.
\]

We now focus on a single-sided dual constraint

\[
\max_{u: \|u\|_2 \leq 1} v^T (X_u)^+ \leq \beta,
\tag{16}
\]

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

\[ P_S := \{ u : x_i^T u \geq 0, \forall i \in S, \ x_i^T u \leq 0, \forall j \in S^c \}. \]

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

\[
\mathcal{H}_X = \bigcup \left\{ \{\text{sign}(X_u)\} : u \in \mathbb{R}^d \right\}.
\]

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

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

\[ S_X := \{ \{\cup_{h_i=1} \{i\}\} : h \in \mathcal{H}_X \}. \]

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

\[
\begin{align*}
\max_{u: \|u\|_2 \leq 1} v^T \text{Diag}(X_u)^+ \\
= \max_{u: \|u\|_2 \leq 1} v^T \text{Diag}(X_u \geq 0)Xu \\
= \max_{S \subseteq [n]} \max_{u: \|u\|_2 \leq 1} v^T \text{Diag}(X_u \geq 0)Xu \\
= \max_{S \subseteq [n]} \max_{u: \|u\|_2 \leq 1} v^T \text{Diag}(X_u \geq 0)Xu \\
= \max_{S \subseteq [n]} \max_{u: \|u\|_2 \leq 1} v^T \text{Diag}(X_u \geq 0)Xu.
\end{align*}
\]

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

\[ P_S = \{ u : D(S)Xu \geq 0, (I_n - D(S))u \leq 0 \}, \]

and the maximization in the dual constraint as

\[
\max_{u: \|u\|_2 \leq 1} v^T (X_u)^+ = \max_{S \subseteq [n]} \max_{u: \|u\|_2 \leq 1} v^T D(S)Xu.
\]

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

\[
\text{min}_{\alpha, \beta \geq 0} \left\{ \sum_{i \in [M]} \sum_{\alpha_i, \beta_i \in \mathbb{R}^n} \text{sign}(X_u)^+ \right\}.
\]

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 order cone constraints as follows

\[
\begin{align*}
\max_{\alpha_i, \beta_i \geq 0} & \sum_{i \in [M]} \sum_{\alpha_i, \beta_i \in \mathbb{R}^n} \text{sign}(X_u)^+ \\
\text{s.t.} & \|X^T D(S_1) (v + \alpha_1 + \beta_1) - X^T \beta_1 \|_2 \leq \beta \\
& \vdots \\
& \|X^T D(S_M) (v + \alpha_M + \beta_M) - X^T \beta_M \|_2 \leq \beta \\
& \|X^T D(S_1) (v + \alpha_1' + \beta_1') - X^T \beta_1' \|_2 \leq \beta \\
& \vdots \\
& \|X^T D(S_M) (v + \alpha_M' + \beta_M') - X^T \beta_M' \|_2 \leq \beta.
\end{align*}
\]

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

See Appendix A.13 for generic convex loss functions.
Neural Networks are Convex Regularizers

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

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

\[
\min_{w_i, \lambda', \beta, \forall \lambda \in \mathbb{R}^M} \max_{v \in \mathbb{R}^n} -\frac{1}{2} \|v - y\|^2 + \frac{1}{2} \|y\|^2 \\
+ \sum_{i=1}^{M} \lambda (\beta + r_i^T X^T D(S_i) (v + \alpha_i + \beta_i) - r_i^T X^T \beta_i) \\
+ \sum_{i=1}^{M} \lambda'_i (\beta + r'^_i X^T D(S_i) (-v + \alpha'_i + \beta'_i) - r'^_i X^T \beta'_i) \\
\]

We note that the objective is concave in \( v, \alpha_i, \beta_i \) 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

\[
\min_{\lambda, \lambda' \in \mathbb{R}^M} \min_{r_i, \forall i} \max_{\alpha, \beta \geq 0, \forall i} -\frac{1}{2} \|v - y\|^2 + \frac{1}{2} \|y\|^2 \\
+ \sum_{i=1}^{M} \lambda (\beta + r_i^T X^T D(S_i) (v + \alpha_i + \beta_i) - r_i^T X^T \beta_i) \\
+ \sum_{i=1}^{M} \lambda'_i (\beta + r'^_i X^T D(S_i) (-v + \alpha'_i + \beta'_i) - r'^_i X^T \beta'_i) \\
\]

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

\[
\min_{\lambda, \lambda' \in \mathbb{R}^M} \min_{r_i, \forall i} \max_{\alpha, \beta \geq 0, \forall i} \left\{ \min_{\alpha, \beta \geq 0, \forall i} \sum_{i=1}^{M} \lambda_i D(S_i) Xr_i \right\} \\
- \lambda'_i D(S_i) Xr'_i - y \right\|_2^2 + \beta \sum_{i=1}^{M} (\lambda_i + \lambda'_i)
\]

Now we apply a change of variables and define \( w_i = \lambda_i r_i \) 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

\[
\min_{w_i, w'_i \in \mathbb{R}^n} \frac{1}{2} \sum_{i=1}^{M} D(S_i) (w_i - w'_i) - y \right\|_2^2 + \beta \sum_{i=1}^{M} (\lambda_i + \lambda'_i)
\]

The variables \( \lambda_i, \lambda'_i, i = 1, \ldots, M \) 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

\[
\min_{w_i, w'_i \in \mathbb{P}^n} \frac{1}{2} \sum_{i=1}^{M} D(S_i) (w_i - w'_i) - y \right\|_2^2 + \beta \sum_{i=1}^{M} (\|w_i\|_2^2 + \|w'_i\|_2^2) 
\]

which is identical to (9), and proves that the objective values are equal. Given a solution to (9), we can form the network output as prescribed in the theorem statement, there will be \( m^* \) pairs \( (v^*_i, w^*_i), \forall i \in [M] \), where either \( v^*_i \) or \( w^*_i \) is non-zero since either the constraint \( v_i^T (X u^*)_+ \leq \beta \), or \( v_i^T (X u^*)_+ \leq \beta \) can be active at any optima in (15). Constructing \( \{v^*_i, w^*_i\}_{i=1}^{m^*} \) as stated in the theorem, and plugging in the non-convex objective (2), we obtain the value

\[
p^* = \frac{1}{2} \sum_{i=1}^{m^*} (X u^*_i + \alpha^*_i - y_i) \right\|_2^2 + \beta \sum_{i=1}^{m^*} \left\{ \frac{1}{2} \left( \frac{\|u^*_i\|_2^2}{\|v^*_i\|_2^2} \right) + \sqrt{\|u^*_i\|_2^2} \right\} 
\]

which is identical to the objective value of the convex program (9). 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 (9) 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. 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, Additional experiments can be found in the Appendix A.1.
Neural Networks are Convex Regularizers

![Graphs](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 the objective value obtained by the proposed convex program in (8).

![Graphs](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.

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 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 $D_1, \ldots, D_P$ which match the sign patterns of the optimized (by GD) 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
Neural Networks are Convex Regularizers

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) Test accuracy

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).

(a) Objective value  
(b) Distance to the solution of the convex program

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 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...
Neural Networks are Convex Regularizers

leveraging convexity, and high dimensional regularization theory \cite{Wainwright2019}. In the light of our results, one can view backpropagation as a heuristic method to solve the convex program \cite{4}, since the global minima are necessarily stationary points of the non-convex objective \cite{2}, i.e., fixed points of the update rule. Efficient optimization algorithms that approximate the convex program can be developed for larger scale experiments.

Acknowledgements

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

References


Neural Networks are Convex Regularizers


