Global Optimality Beyond Two Layers: Training Deep ReLU Networks via Convex Programs

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

Abstract

Understanding the fundamental mechanism behind the success of deep neural networks is one of the key challenges in the modern machine learning literature. Despite numerous attempts, a solid theoretical analysis is yet to be developed. In this paper, we develop a novel unified framework to reveal a hidden regularization mechanism through the lens of convex optimization. We first show that the training of multiple three-layer ReLU sub-networks with weight decay regularization can be equivalently cast as a convex optimization problem in a higher dimensional space, where sparsity is enforced via a group ℓ_1-norm regularization. Consequently, ReLU networks can be interpreted as high dimensional feature selection methods. More importantly, we then prove that the equivalent convex problem can be globally optimized by a standard convex optimization solver with a polynomial-time complexity with respect to the number of samples and data dimension when the width of the network is fixed. Finally, we numerically validate our theoretical results via experiments involving both synthetic and real datasets.

1. Introduction

Deep neural networks have been extensively studied in machine learning, natural language processing, computer vision, and many other fields. Even though deep neural networks have provided dramatic improvements over conventional learning algorithms (Goodfellow et al., 2016), fundamental mathematical mechanisms behind this success remain elusive. To this end, in this paper, we investigate the implicit mechanisms behind deep neural networks by leveraging tools available in convex optimization theory.

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

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

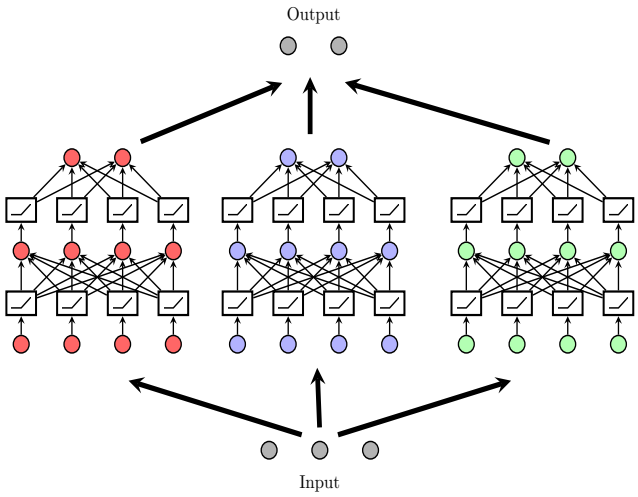
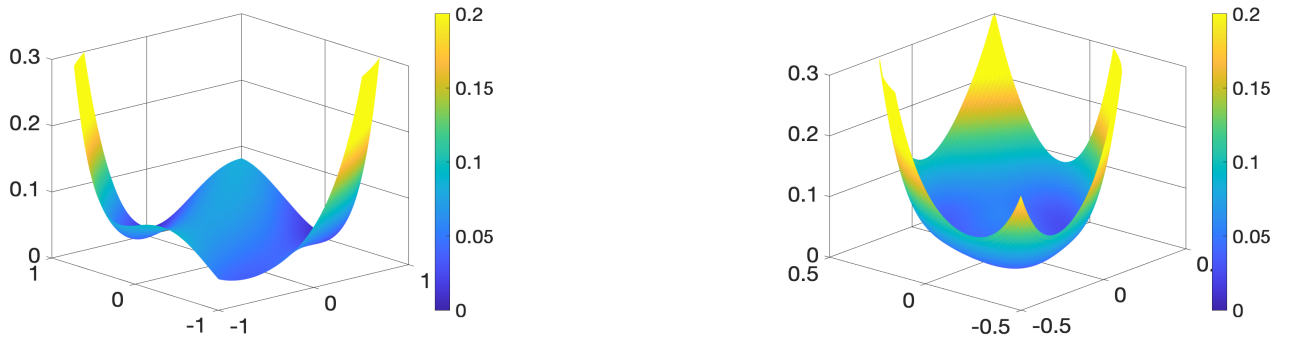


Figure 1: An architecture with three sub-networks, i.e., $K = 3$, where each is a three-layer ReLU network.

1.1. Related Work

The problem of learning the parameters of a deep neural network is the subject of many studies due to its highly non-convex and nonlinear nature. Despite their complex structure, deep networks are usually trained by simple local search algorithms such as Gradient Descent (GD) to achieve a globally optimal set of parameters (Brutzkus & Globerson, 2017). However, it has been shown that these algorithms might also converge a local minimum in pathological cases (Shalev-Shwartz et al., 2017; Goodfellow et al., 2016; Ergen & Pilanci, 2019). In addition to this, (Ge et al., 2017; Safran & Shamir, 2018) proved that Stochastic GD (SGD) is likely to get stuck at a local minimum when the number of parameters is small. Furthermore, (Anandkumar & Ge, 2016) reported that complicated saddle points do exist in the optimization landscape of deep networks. Since such points might be hard to escape for local search algorithms, training deep neural networks is a computationally challenging optimization problem (DasGupta et al., 1995; Blum & Rivest, 1989; Bartlett & Ben-David, 1999).



(a) $K = 1$ (b) $K = 50$

Figure 2: The loss landscape of an architecture consisting of K three-layer ReLU networks trained on a synthetic dataset, where the number of samples, features, and neurons are chosen as $(n, d, m_1) = (100, 10, 10)$, respectively. For the dataset, we first randomly generate a data matrix by sampling a multivariate standard normal distribution and then obtain the corresponding label vector by feeding the data matrix into a randomly initialized teacher network with $m_1 = 10$.

In order to alleviate these training issues, a line of research focused on designing new architectures that enjoy a well behaved optimization landscape by leveraging over-parameterization (Brutzkus et al., 2017; Du & Lee, 2018; Arora et al., 2018; Neyshabur et al., 2018) and combining multiple neural network architectures, termed as sub-networks, in parallel (Iandola et al., 2016; Szegedy et al., 2017; Chollet, 2017; Xie et al., 2017; Zagoruyko & Komodakis, 2016; Veit et al., 2016) as illustrated in Figure 1. Such studies empirically proved that increasing the number of sub-networks yields less complicated optimization landscapes so that GD generally converges to a global minimum. These empirical observations are due to the fact that increasing the number of sub-networks yields a less non-convex optimization landscape. To support this claim, we also provide an experiment in Figure 2, where increasing the number of sub-network clearly promotes convexity of the loss landscape. In addition to better training performance, neural network architectures with multiple sub-networks also enjoy a remarkable generalization performance so that various such architectures have been introduced to achieve state-of-the-art performance in practice, especially for image classification tasks. As an example, SqueezeNet (Iandola et al., 2016), Inception (Szegedy et al., 2017), Xception (Chollet, 2017), and ResNext (Xie et al., 2017) are combinations of multiple networks and achieved notable improvements in practice.

1.2. Our Contributions

Our contributions can be summarized as follows:

- We introduce an exact analytical framework based on convex duality to characterize the optimal solutions to regularized deep ReLU network training problems. As a corollary, we provide interpretations for the convergence of local search algorithms such as SGD and the loss landscape of these training problems. We also numerically verify these interpretations via experiments involving both synthetic and real benchmark datasets.

- We show that the training problem of an architecture with multiple ReLU sub-networks can be equivalently stated as a convex optimization problem. More importantly, we prove that the equivalent convex problem can be globally optimized in polynomial-time using standard convex optimization solvers. Therefore, we prove the polynomial-time trainability of regularized ReLU networks with multiple nonlinear layers, which generalizes the recent two-layer results in (Pilanci & Ergen, 2020; Ergen & Pilanci, 2020a;b) to a much broader class of neural network architectures.

- Our analysis also reveals an implicit regularization structure behind the non-convex ReLU network training problems. In particular, we show that this implicit regularization is a group ℓ_1-norm regularization, which encourages sparsity for the equivalent convex problem in a high dimensional space. We further prove that there is a direct mapping between the original non-convex and the equivalent convex training problems so that sparsity for the convex problem implies a smaller number of sub-networks for the original non-convex problem.

- Unlike the previous studies, our results hold for arbitrary convex loss functions including squared, cross entropy, and hinge loss, and common regularization methods, e.g., weight decay.

1.3. Notation

We denote matrices and vectors as uppercase and lowercase bold letters, respectively. We use $\mathbf{I}_k$ to denote the identity matrix of size $k \times k$ and $\mathbf{0}$ (or $\mathbf{1}$) to denote a vector/matrix of zeros (or ones) with appropriate sizes. We denote the set of integers from 1 to n as $[n]$. In addition, $\|\cdot\|_2$ and $\|\cdot\|_F$ denotes the Euclidean and Frobenius norms, respectively. Furthermore, we define the unit ℓ_p-ball as $\mathcal{B}_p := \{\mathbf{u} \in \mathbb{R}^d : \|\mathbf{u}\|_p \leq 1\}$. We also use $\mathbf{1}[x \geq 0]$ and $(x)_+ = \max\{x, 0\}$ to denote the elementwise 0-1 valued indicator and ReLU, respectively.

1.4. Overview of Our Results

In this paper, we consider an architecture with K sub-networks each of which is an L-layer ReLU network with layer weights $\mathbf{W}_{lk} \in \mathbb{R}^{m_{l-1} \times m_l}$, $\forall l \in [L]$, where $m_0 = d$, $m_L = 1$[1], and m_l denotes the number of neurons in the l^{th} hidden layer. Given an input data matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ and the corresponding label vector $\mathbf{y} \in \mathbb{R}^n$, the regularized network training problem can be formulated as follows

$$\min_{\theta \in \Theta} \mathcal{L}\left(\sum_{k=1}^{K} f_{\theta,k}(\mathbf{X}), \mathbf{y}\right) + \beta \sum_{k=1}^{K} \mathcal{R}_k(\theta)\,, \qquad (1)$$

where Θ is the parameter space, $\mathcal{L}(\cdot,\cdot)$ is an arbitrary convex loss function, $\mathcal{R}_k(\cdot)$ is the regularization function for the layer weights in the k^{th} sub-network, and $\beta > 0$ is a regularization parameter. In addition to this, we compactly define the set of network parameters as $\theta := \{\{\mathbf{W}_{lk}\}_{l=1}^{L}\}_{k=1}^{K}$, $\theta \in \Theta$ and the output of each sub-network as

$$f_{\theta,k}(\mathbf{X}) := \left((\mathbf{X}\mathbf{W}_{1k})_+ \ldots \mathbf{w}_{(L-1)k}\right)_+ w_{Lk}.$$

Remark 1. *Notice that the function parallel architectures include a wide range of neural network architectures in practice. As an example, ResNets (He et al., 2016) are a special case of this architecture. We first note that residual blocks are applied after ReLU in practice so that the input to each block has nonnegative entries. Hence, for this special case, we assume $\mathbf{X} \in \mathbb{R}_+^{n \times d}$. Let us consider a four-layer architecture with $K = 2$, $\mathbf{W}_{11} = \mathbf{W}_1$, $\mathbf{W}_{21} = \mathbf{W}_2$, $\mathbf{W}_{12} = \mathbf{W}_{22} = \mathbf{I}_d$, $\mathbf{w}_{31} = \mathbf{w}_{32} = \mathbf{w}_3$, and $w_{41} = w_{42} = w_4$ then*

$$\begin{aligned} f_{\theta,k}(\mathbf{X}) &= \sum_{k=1}^{2} \left(\left((\mathbf{X}\mathbf{W}_{1k})_+ \mathbf{W}_{2k}\right)_+ \mathbf{w}_{3k}\right)_+ w_{4k} \\ &= \left(\left((\mathbf{X}\mathbf{W}_1)_+ \mathbf{W}_2\right)_+ \mathbf{w}_3\right)_+ w_4 + (\mathbf{X}\mathbf{w}_3)_+ w_4 \end{aligned}$$

which corresponds to a shallow ResNet as depicted in Figure 1 of (Veit et al., 2016).

Throughout the paper, we consider the conventional regression framework with weight decay regularization and squared loss, i.e., $\mathcal{L}(f_\theta(\mathbf{X}), \mathbf{y}) = \|f_\theta(\mathbf{X}) - \mathbf{y}\|_2^2$ and $\mathcal{R}(\theta) = \frac{1}{2}\|\theta\|_2^2$. However, our derivations also hold for arbitrary convex loss functions including hinge loss and cross entropy and vector outputs as proven in the supplementary file. Thus, we consider the following optimization problem

$$P^* := \min_{\theta \in \Theta} \mathcal{L}\left(\sum_{k=1}^{K} f_{\theta,k}(\mathbf{X}), \mathbf{y}\right) + \frac{\beta}{2} \sum_{k=1}^{K} \sum_{l=L-1}^{L} \|\mathbf{W}_{lk}\|_F^2, \qquad (2)$$

where $\Theta := \{\theta : \|\mathbf{W}_{lk}\|_F \leq 1, \forall l \in [L-2], \forall k \in [K]\}$ without loss of generality.

Remark 2. *In (2), we impose unit Frobenius norm constraints on the first $L-2$ layer weights and regularize only the last two layers. Although this might appear to limit the effectiveness of the regularization on the network output, in Lemma 1, we show that as long as the last two layers' weights of each sub-network are regularized, the remaining layer weights do not change the structure of the regularization. They only contribute to the ratio between the training error and regularization term. Therefore, one can undo this by simply tuning β (see Section A.3 of the supplementary file for details).*

Next, we introduce a rescaling technique to equivalently state the problem in (2) as an ℓ_1-norm minimization problem, which is critical for strong duality to hold.

Lemma 1. *The following problems are equivalent* [2]*:*

$$\begin{aligned} &\min_{\theta \in \Theta} \mathcal{L}\left(\sum_{k=1}^{K} f_{\theta,k}(\mathbf{X}), \mathbf{y}\right) + \frac{\beta}{2} \sum_{k=1}^{K} \sum_{l=L-1}^{L} \|\mathbf{W}_{lk}\|_F^2 \\ &= \min_{\theta \in \Theta_p} \mathcal{L}\left(\sum_{k=1}^{K} f_{\theta,k}(\mathbf{X}), \mathbf{y}\right) + \beta \sum_{k=1}^{K} |w_{Lk}|, \end{aligned}$$

where $\Theta_p := \{\theta \in \Theta : \|\mathbf{W}_{lk}\|_F \leq 1, \forall l \in [L-2],\ \|\mathbf{w}_{(L-1)k}\|_2 \leq 1, \forall k \in [K]\}$.

Using Lemma 1, we first take the dual of ℓ_1 equivalent of (2) with respect to the output weights w_{Lk} and then change the order of min-max to achieve the following dual problem, which provides a lower bound for the primal problem (2)[3]

$$\begin{aligned} &P^* \geq D^* := \max_{\mathbf{v}} -\mathcal{L}^*(\mathbf{v}) \qquad (3) \\ &\text{s.t. } \max_{\theta \in \Theta_p} \left|\mathbf{v}^T \left((\mathbf{X}\mathbf{W}_1)_+ \ldots \mathbf{w}_{(L-1)}\right)_+\right| \leq \beta, \end{aligned}$$

[1] We consider scalar output networks for presentation simplicity, however, all our derivations can be extended to vector output networks as shown in Section A.7 of the supplementary file.

[2] All the proofs are presented in the supplementary file.

[3] We present the details in Section A.6 of the supplementary file.

where $\mathcal{L}^*$ is the Fenchel conjugate function defined as (Boyd & Vandenberghe, 2004)

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

Using this dual characterization, we first find a set of hidden layer weights via the optimality conditions (i.e. active dual constraints). We then prove the optimality of these weights via strong duality, i.e., $P^* = D^*$.

1.5. Prior Work (Zhang et al., 2019; Haeffele & Vidal, 2017; Pilanci & Ergen, 2020)

Here, we further clarify contributions and limitations of some recent studies (Zhang et al., 2019; Haeffele & Vidal, 2017; Pilanci & Ergen, 2020) that focus on the training problem of architectures with sub-networks through the lens of convex optimization theory. (Haeffele & Vidal, 2017) particularly analyzed the characteristics of the local minima of the regularized training objective in (1). However, the results are valid under several impractical assumptions. As an example, they require all local minima of (1) to be rank-deficient. Additionally, they assume that the objective function (1) is twice differentiable, which is not the case for non-smooth problems, e.g., training problems with ReLU activation. Furthermore, they require K to be too large to be of practical use. Finally, their proof techniques depend on finding a local descent direction of a non-convex training problem, which might be NP hard in general. Therefore, even though this study provided valuable insights for future research, it is far from explaining observations in practical scenarios.

In addition to (Haeffele & Vidal, 2017), (Zhang et al., 2019) provided some results on strong duality. They particularly showed that the primal-dual gap diminishes as the number of sub-networks K increases. Although this study is an important step to understand deep networks through convex duality, it does not include any solid results for finite-size networks. Moreover, their results require strict assumptions: 1) the analysis only works for hinge loss and linear networks; 2) the analysis requires the data matrix to be included in the regularization term $\mathcal{R}(\theta)$, thus, it is not valid for commonly used regularizations such as weight decay in (2); 3) they require assumptions on the regularization parameter β.

Another closely related work (Pilanci & Ergen, 2020) studied convex optimization for ReLU networks and followed by a series of papers (Sahiner et al., 2021; Ergen & Pilanci, 2021; Ergen et al., 2021; Gupta et al., 2021). Particularly, the authors introduced an exact convex formulation to train two-layer ReLU networks in polynomial-time for training data $\mathbf{X} \in \mathbb{R}^{n\times d}$ of constant rank, where the network output is $f_\theta(\mathbf{X}) := \sum_{j=1}^m \left(\mathbf{X}\mathbf{u}_j\right)_+ \alpha_j$ given the hidden layer weights $\mathbf{u}_j \in \mathbb{R}^d$ and the output layer weights $\alpha_j \in \mathbb{R}$. However, their analysis does not extend to deep architectures with more than one ReLU layer. The reason for this limitation is that the composition of multiple ReLU layers is a significantly more challenging optimization problem. Moreover, in such a case, the complexity becomes exponential-time due to the complex and combinatorial behavior of multiple ReLU layers.

2. Architectures with Three-layer ReLU Sub-networks

Here, we consider K three-layer ReLU sub-networks, i.e., $L = 3$, trained with squared loss. Given a dataset $\{\mathbf{X}, \mathbf{y}\}$, the regularized training problem is as follows

$$P^* = \min_{\theta\in\Theta} \frac{1}{2} \left\|f_\theta(\mathbf{X}) - \mathbf{y}\right\|_2^2 + \frac{\beta}{2} \sum_{k=1}^K \left(\|\mathbf{w}_{2k}\|_2^2 + w_{3k}^2\right), \tag{4}$$

where $\Theta := \{\theta : \|\mathbf{W}_{1k}\|_F \leq 1,\ \forall k \in [K]\}$ and

$$f_\theta(\mathbf{X}) := \sum_{k=1}^K f_{\theta,k}(\mathbf{X}).$$

Using the rescaling in Lemma 1, (4) can be written as

$$P^* = \min_{\theta\in\Theta_p} \frac{1}{2} \left\|f_\theta(\mathbf{X}) - \mathbf{y}\right\|_2^2 + \beta\|\mathbf{w}_3\|_1. \tag{5}$$

We then take the dual with respect to $\mathbf{w}_3$ and then change the order of min-max to obtain the following dual problem

$$\begin{aligned} P^* \geq D^* := \max_{\mathbf{v}} & -\frac{1}{2}\|\mathbf{v} - \mathbf{y}\|_2^2 + \frac{1}{2}\|\mathbf{y}\|_2^2 \\ \text{s.t. } & \max_{\theta\in\Theta_p} \left|\mathbf{v}^T \left(\left(\mathbf{X}\mathbf{W}_1\right)_+ \mathbf{w}_2\right)_+\right| \leq \beta. \end{aligned} \tag{6}$$

In order to obtain the bidual of (4), we again take the dual of (6) with respect to $\mathbf{v}$, which yields

$$\begin{aligned} P_B^* := \min_{\boldsymbol{\mu}} \frac{1}{2} \left\| \int_{\theta\in\Theta_p} \left(\left(\mathbf{X}\mathbf{W}_1\right)_+ \mathbf{w}_2\right)_+ d\mu(\theta) - \mathbf{y} \right\|_2^2 \\ + \beta\|\boldsymbol{\mu}\|_{TV}, \end{aligned} \tag{7}$$

where $\|\boldsymbol{\mu}\|_{TV}$ is the total variation norm of the Radon measure $\boldsymbol{\mu}$. We now note that (7) is an infinite-size regularized neural network training problem studied in (Bach, 2017), and it is convex. Therefore, strong duality holds, i.e., $D^* = P_B^*$. We also note that although (7) involves an infinite dimensional integral, by Caratheodory's theorem, this integral form can be represented as a finite summation of at most $n + 1$ Dirac delta measures (Rosset et al., 2007). Thus, we select $\boldsymbol{\mu}' = \sum_{i=1}^{K^*} w_{3i}\, \delta(\theta - \theta_i)$, where $K^* \leq n + 1$, to achieve the following finite-size problem

$$P_B^* = \min_{\theta\in\Theta_p} \frac{1}{2} \left\| \sum_{i=1}^{K^*} f_{\theta,i}(\mathbf{X}) - \mathbf{y} \right\|_2^2 + \beta\|\mathbf{w}_3\|_1. \tag{8}$$

Note that (8) is the same problem with (5) provided that $K \geq K^*$. Therefore, strong duality holds, i.e., $P^* = P_B^* = D^*$. Using strong duality, we first characterize optimal hidden layer weights via the active constraints of the dual problem. We then introduce a novel framework to represent the constraints in a convex form to obtain an equivalent convex formulation for the primal problem (4).

We now represent the constraint in (6) as

$$\left\{\mathbf{v} : \max_{\theta \in \Theta_p} \mathbf{v}^T \left(\left(\mathbf{X}\mathbf{W}_1\right)_+ \mathbf{w}_2\right)_+ \leq \beta\right\} \bigcap \left\{\mathbf{v} : \max_{\theta \in \Theta_p} -\mathbf{v}^T \left(\left(\mathbf{X}\mathbf{W}_1\right)_+ \mathbf{w}_2\right)_+ \leq \beta\right\}.$$

We first focus on a single-sided dual constraint

$$\max_{\theta \in \Theta_p} \mathbf{v}^T \left(\left(\mathbf{X}\mathbf{W}_1\right)_+ \mathbf{w}_2\right)_+ \leq \beta. \tag{9}$$

Noting that $\Theta_p = \{\theta \in \Theta : \|\mathbf{W}_{1k}\|_F \leq 1,\ \|\mathbf{w}_{2k}\|_2 \leq 1, \forall k \in [K]\}$, we then equivalently write the constraint in (9) as

$$\max_{\mathcal{I}_j \in \{\pm 1\}} \max_{\substack{\theta \in \Theta_p \\ \mathbf{w}_2 \geq 0}} \mathbf{v}^T \left(\sum_{j=1}^{m_1} \mathcal{I}_j \left(\mathbf{X}\mathbf{w}_{1j} w_{2j}\right)_+\right)_+ \leq \beta, \tag{10}$$

where $\mathcal{I}_j = \text{sign}(w_{2j}) \in \{+1, -1\}$. Now, modifying $\|\mathbf{W}_1\|_F \leq 1$ as $\|\mathbf{w}_{1j}\|_2^2 \leq t_j$ such that $\mathbf{1}^T\mathbf{t} \leq 1$ and defining $\mathbf{w}'_{1j} = \sqrt{w'_{2j}}\mathbf{w}_{1j}$, where $w'_{2j} = w_{2j}^2$, yield

$$\max_{\substack{t_j \geq 0 \\ \mathbf{1}^T\mathbf{t} \leq 1 \\ \mathcal{I}_j \in \{\pm 1\}}} \max_{\substack{\|\mathbf{w}'_{1j}\|_2^2 / w'_{2j} \leq t_j \\ \mathbf{1}^T \mathbf{w}'_2 \leq 1 \\ \mathbf{w}'_2 \geq 0}} \mathbf{v}^T \left(\sum_{j=1}^{m_1} \mathcal{I}_j \left(\mathbf{X}\mathbf{w}'_{1j}\right)_+\right)_+ \leq \beta. \tag{11}$$

We remark that (11) is non-convex due to the ReLU activation. Therefore, to eliminate ReLU without altering the constraints, we introduce a notion of *hyperplane arrangement* as follows.

Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be the sets of all hyperplane arrangements for the hidden layers, which are defined as

$$\mathcal{H}_1 := \bigcup \left\{\{\text{sign}(\mathbf{X}\mathbf{w})\} : \mathbf{w} \in \mathbb{R}^d\right\}$$
$$\mathcal{H}_2 := \bigcup \left\{\{\text{sign}(\left(\mathbf{X}\mathbf{W}_1\right)_+ \mathbf{w}_2)\} : \mathbf{W}_1 \in \mathbb{R}^{d \times m_1}, \mathbf{w}_2 \in \mathbb{R}^{m_1}\right\}.$$

We next define an alternative representation of the sign patterns in $\mathcal{H}_1$ and $\mathcal{H}_2$, which is the collection of sets that correspond to positive signs for each element in $\mathcal{H}_j$ as follows

$$\mathcal{S}_j := \left\{\{\cup_{h_i=1}\{i\}\} : \mathbf{h} \in \mathcal{H}_j\right\}, \forall j \in [2].$$

We note that ReLU is an elementwise function that masks the negative entries of a vector/matrix. Hence, we define two diagonal mask matrices $\mathbf{D}(S_j) \in \mathbb{R}^{n \times n}$ as $\mathbf{D}(S_j)_{ii} := \mathbf{1}[i \in S_j]$. We now enumerate all hyperplane arrangements and signs, and index them in an arbitrary order, which are denoted as $\mathcal{I}_j$, $\mathbf{D}_{1ij}$, and $\mathbf{D}_{2l}$, where $i \in [P_1]$, $l \in [P_2]$, $P_1 = |\mathcal{S}_1|$, and $P_2 = |\mathcal{S}_2|$. We then rewrite (11) as

$$\max_{\substack{i \in [P_1] \\ l \in [P_2]}} \max_{\substack{t_j \geq 0 \\ \mathbf{1}^T\mathbf{t} \leq 1 \\ \mathcal{I}_j \in \{\pm 1\}}} \max_{\substack{\|\mathbf{w}'_{1j}\|_2^2 / w'_{2j} \leq t_j \\ \mathbf{1}^T \mathbf{w}'_2 \leq 1 \\ \mathbf{w}'_2 \geq 0}} \mathbf{v}^T \mathbf{D}_{2l} \sum_{j=1}^{m_1} \mathcal{I}_j \mathbf{D}_{1ij} \mathbf{X}\mathbf{w}'_{1j}$$
$$\text{s.t. } (2\mathbf{D}_{1ij}) - \mathbf{I}_n)\mathbf{X}\mathbf{w}'_{1j} \geq 0,\ \forall i, j,$$
$$(2\mathbf{D}_{2l} - \mathbf{I}_n) \sum_{j=1}^{m_1} \mathcal{I}_j \mathbf{D}_{1ij} \mathbf{X}\mathbf{w}'_{1j} \geq 0,\ \forall i, l,$$

where we use an alternative representation for ReLU as $\left(\mathbf{X}\mathbf{w}_1\right)_+ = \mathbf{D}\mathbf{X}\mathbf{w}_1$ provided that $(2\mathbf{D} - \mathbf{I}_n)\mathbf{X}\mathbf{w}_1 \geq 0$. Therefore, we can convert the non-convex dual constraints in (9) to a convex constraint given fixed diagonal matrices $\{\mathbf{D}_{1ij}\}_{j=1}^{m_1}$, $\mathbf{D}_{2l}$ and a fixed set of signs $\{\mathcal{I}_j\}_{j=1}^{m_1}$ (see Section A.4 for details).

Using this new representation for the dual constraints, we then take the dual of (6) to obtain the convex bidual form of the primal problem (4) as described in the next theorem.

Theorem 1. *The non-convex training problem in* (4) *can be equivalently stated as a convex problem as follows*

$$\min_{\mathbf{w}, \mathbf{w}' \in \mathcal{C}} \frac{1}{2} \left\|\tilde{\mathbf{X}}\left(\mathbf{w}' - \mathbf{w}\right) - \mathbf{y}\right\|_2^2 + \beta \left(\|\mathbf{w}\|_{2,1} + \|\mathbf{w}'\|_{2,1}\right) \tag{12}$$

where $\|\cdot\|_{2,1}$ *is* d *dimensional group norm operator such that given a vector* $\mathbf{u} \in \mathbb{R}^{dP}$, $\|\mathbf{u}\|_{2,1} := \sum_{i=1}^{P} \|\mathbf{u}_i\|_2$, *where* $\mathbf{u}_i$*'s are the ordered* d *dimensional partitions of* $\mathbf{u}$. *Moreover,* $\tilde{\mathbf{X}} \in \mathbb{R}^{n \times 2dm_1P_1P_2}$ *and* $\mathcal{C}$ *are defined as*

$$\mathcal{C} := \Big\{\mathbf{w} \in \mathbb{R}^{2dm_1P_1P_2} :$$
$$(2\mathbf{D}_{1ij} - \mathbf{I}_n)\mathbf{X}\mathbf{w}^+_{ijl} \geq 0,\ (2\mathbf{D}_{1ij} - \mathbf{I}_n)\mathbf{X}\mathbf{w}^-_{ijl} \leq 0,$$
$$(2\mathbf{D}_{2l} - \mathbf{I}_n) \sum_{j=1}^{m_1} \mathbf{D}_{1ij}\mathbf{X}\mathbf{w}^{\pm}_{ijl} \geq 0, \forall i, j, l, \pm\Big\}$$
$$\tilde{\mathbf{X}} := \begin{bmatrix} \tilde{\mathbf{X}}_s & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{X}}_s \end{bmatrix},$$

where $\mathbf{w}, \mathbf{w}' \in \mathbb{R}^{2dm_1P_1P_2}$ *are the vectors constructed by concatenating* $\{\{\{\{\mathbf{w}^{\pm}_{ijl}\}_{i=1}^{P_1}\}_{j=1}^{m_1}\}_{l=1}^{P_2}\}_{\pm}$ *and* $\{\{\{\{\mathbf{w}^{\pm'}_{ijl}\}_{i=1}^{P_1}\}_{j=1}^{m_1}\}_{l=1}^{P_2}\}_{\pm}$, *respectively, and*

$$\tilde{\mathbf{X}}_s := \left[\mathbf{D}_{21}\mathbf{D}_{111}\mathbf{X} \ \ldots \ \mathbf{D}_{2l}\mathbf{D}_{1ij}\mathbf{X} \ \ldots \ \mathbf{D}_{2P_2}\mathbf{D}_{1P_1m_1}\mathbf{X}\right].$$

Difference between convex programs for two-layer (Pilanci & Ergen, 2020) and three-layer (ours) networks: (Pilanci & Ergen, 2020) introduced a convex program for two-layer networks. In that case, since there is only a single ReLU layer, the data matrix $\mathbf{X}$ is multiplied with a single hyperplane arrangement matrix, namely $\mathbf{D}_i$, and the effective high dimensional data matrix becomes $\tilde{\mathbf{X}}_s = [\mathbf{D}_1\mathbf{X} \ldots \mathbf{D}_P\mathbf{X}]$. However, our architecture in (4) has two ReLU layers, combination of which can generate significantly more complex features, which are associated with local variables $\{\mathbf{w}_{ijl}^{\pm}\}$ that interact with data through the multiplication of two diagonal matrices as $\tilde{\mathbf{X}}_s = [\mathbf{D}_{21}\mathbf{D}_{111}\mathbf{X} \ldots \mathbf{D}_{2P_2}\mathbf{D}_{1P_1m_1}\mathbf{X}]$. In particular, a deep network can be precisely interpreted as a high-dimensional feature selection method due to convex group sparsity regularization, which encourages a parsimonious model. In simpler terms, such deep ReLU networks are group lasso models with additional linear constraints. Therefore, our result reveals the impact of having additional layers and its implications on the expressive power of a network.

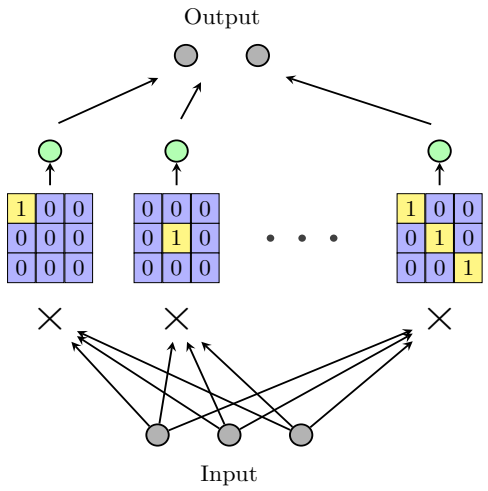


Figure 3: Equivalent convex model for the network in Figure 1. Here, nonlinear ReLU layers are replaced with a linear layer and a transformation that maps to the input data to higher dimensional space by multiplying it with fixed diagonal matrices as detailed in Theorem 1.

The next result shows that there is a direct mapping between the solutions to (4) and (12). Therefore, once we solve the convex program in (12), one can construct the optimal network parameters in their original form in (4) (see Section A.5 for the explicit definitions of the mapping).

Proposition 1. *An optimal solution to the non-convex training problem in* (4)*, i.e.,* $\{\mathbf{W}_{1k}^*, \mathbf{w}_{2k}^*, w_{3k}^*\}_{k=1}^K$*, can be constructed using the optimal solution to* (12)*, denoted as* $(\mathbf{w}^*, \mathbf{w}'^*)$*. Therefore, there is a direct mapping between the architectures in Figure 1 and 3.*

Remark 3. *The problem in* (12) *can be approximated by sampling a set of diagonal matrices* $\{\{\mathbf{D}_{1ij}\}_{i=1}^{\bar{P}_1}\}_{j=1}^{m_1}$ *and* $\{\mathbf{D}_{2l}\}_{l=1}^{\bar{P}_2}$*. As an example, one can generate random vectors* $\mathbf{u}_{ij}$*'s from an arbitrary distribution, e.g.,* $\mathbf{u}_{ij} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$*,* $\bar{P}_1$ *times and then let* $\mathbf{D}_{1ij} = diag(\mathbf{1}[\mathbf{X}\mathbf{u}_{ij} \geq 0]), \forall i \in [\bar{P}_1]$*,* $\forall j \in [m_1]$*. Similarly, one can randomly generate* $\mathbf{U}_1 \in \mathbb{R}^{d \times m_1}$ *and* $\mathbf{u}_2 \in \mathbb{R}^{m_1}$ $\bar{P}_2$ *times and then let* $\mathbf{D}_{2l} = diag\left(\mathbf{1}\left[(\mathbf{X}\mathbf{U}_{1l})_+ \mathbf{u}_{2l} \geq 0\right]\right), \forall l \in [\bar{P}_2]$*. Then, one can solve the convex problem in* (12) *using these hyperplane arrangements. In fact, SGD applied to the non-convex problem in* (4) *can be viewed as an active set optimization strategy to solve the equivalent convex problem, which maintains a small active support. We also note that global optimums of the convex problem are the fixed points for SGD, i.e., stationary points of* (4)*. Furthermore, one can bound the suboptimality of any solution found by SGD for the non-convex problem using the dual of* (12)*.*

2.1. Multilayer Hyperplane Arrangements

It is known that the number of hyperplane arrangements for the first layer, i.e., P_1, can be upper-bounded as follows

$$P_1 \leq 2\sum_{k=0}^{r-1}\binom{n-1}{k} \leq 2r\left(\frac{e(n-1)}{r}\right)^r = \mathcal{O}(n^r) \tag{13}$$

for $r \leq n$, where $r := \text{rank}(\mathbf{X}) \leq d$ (Ojha, 2000; Stanley et al., 2004; Winder, 1966; Cover, 1965). For the second layer, we first note that the activations before ReLU can be written as follows

$$\sum_{j=1}^{m_1}(\mathbf{X}\mathbf{w}_{1j})_+ w_{2j} = \sum_{j=1}^{m_1}\left(\mathbf{X}\mathbf{w}_j'\right)_+ \mathcal{I}_j = \sum_{j=1}^{m_1}\mathcal{I}_j\mathbf{D}_{1j}\mathbf{X}\mathbf{w}_j',$$

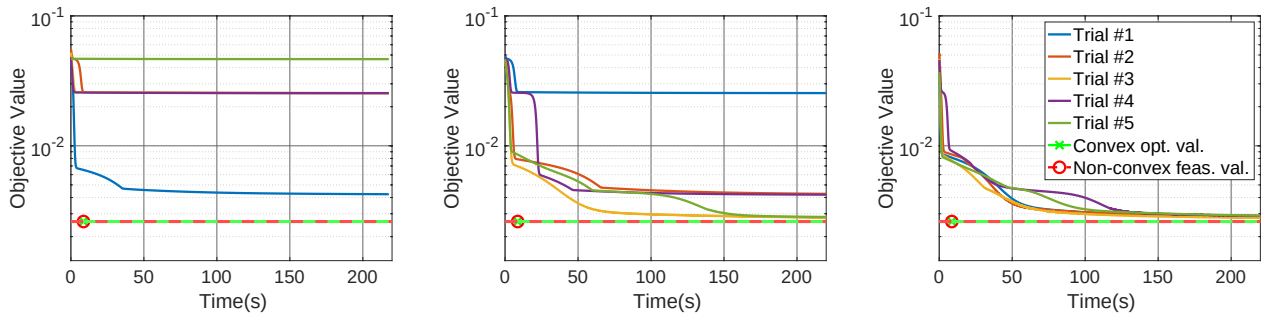
which can also be formulated as a matrix-vector product

$$\underbrace{\left[\mathcal{I}_1\mathbf{D}_{11}\mathbf{X} \quad \mathcal{I}_2\mathbf{D}_{12}\mathbf{X} \quad \ldots \quad \mathcal{I}_{m_1}\mathbf{D}_{1m_1}\mathbf{X}\right]}_{\mathbf{X}'}\underbrace{\text{vec}(\{\mathbf{w}_j'\}_{j=1}^{m_1})}_{\mathbf{w}'}$$
$$= \mathbf{X}'\mathbf{w}',$$

where $\mathbf{X}' \in \mathbb{R}^{n \times m_1 d}$ and $\mathbf{w}' \in \mathbb{R}^{m_1 d}$. Therefore, given a fixed set $\{\mathcal{I}_j, \mathbf{D}_{1j}\}_{j=1}^{m_1}$, the number of hyperplane arrangements for $\mathbf{X}'$ can be upper-bounded as follows

$$P_2' \leq 2r'\left(\frac{e(n-1)}{r'}\right)^{r'} \leq 2m_1 r\left(\frac{e(n-1)}{m_1 r}\right)^{m_1 r},$$

where $r' := \text{rank}(\mathbf{X}') \leq m_1 r$ since $\text{rank}(\mathbf{X}) = r$ and we assume that $m_1 r \leq n$. Since there exist 2 and P_1 possible choices for each $\mathcal{I}_j$ and $\mathbf{D}_{1j}$, respectively, the total number of hyperplane arrangements for the second layer can be



(a) $K = 2$ (b) $K = 5$ (c) $K = 15$

Figure 4: Training cost of a three-layer architecture trained with SGD (5 initialization trials) on a synthetic dataset with $(n, d, m_1, \beta, \text{batch size}) = (5, 2, 3, 0.002, 5)$, where the green line with a marker represents the objective value obtained by the proposed convex program in (12) and the red line with a marker represents the non-convex objective value in (4) of a classical ReLU network constructed from the solution of convex program as described in Proposition 1. Here, we use markers to denote the total computation time of the convex optimization solver.

	Dataset Size		**Training Objective**		**Test Error**	
	n	d	SGD	Convex	SGD	Convex
acute-inflamation (Czerniak & Zarzycki, 2003)	120	6	0.0029	**0.0013**	0.0224	**0.0217**
acute-nephritis (Czerniak & Zarzycki, 2003)	120	6	0.0039	**0.0021**	0.0198	**0.0192**
balloons (Dua & Graff, 2017)	16	4	0.7901	**0.6695**	0.2693	**0.1496**
breast-tissue (Dua & Graff, 2017)	106	9	0.5219	**0.3979**	1.4082	**1.0377**
fertiliy (Dua & Graff, 2017)	100	9	0.125	**0.1224**	**0.3551**	0.5050
pittsburg-bridges-span (Dua & Graff, 2017)	92	7	0.1723	**0.1668**	1.4373	**1.3112**

Table 1: Training objective and test error of a three-layer architecture trained using SGD and the convex program in (12), i.e., denoted as "Convex", on some small scale UCI datasets with $(m_1, K, \beta, \text{batch size}) = (7, 10, 0.1, \lfloor n/8 \rfloor)$.

upper-bounded as follows

$$P_2 \leq P_2'(2P_1)^{m_1} \leq m_1 r 2^{m_1+1} P_1^{m_1} \left(\frac{e(n-1)}{m_1 r}\right)^{m_1 r} \leq \frac{2^{2m_1+1}(e(n-1))^{2m_1 r}}{m_1^{m_1 r-1} r^{2m_1 r-m_1-1}} = \mathcal{O}(n^{m_1 r}), \tag{14}$$

which is polynomial in n and d since m_1 and r are fixed scalars.

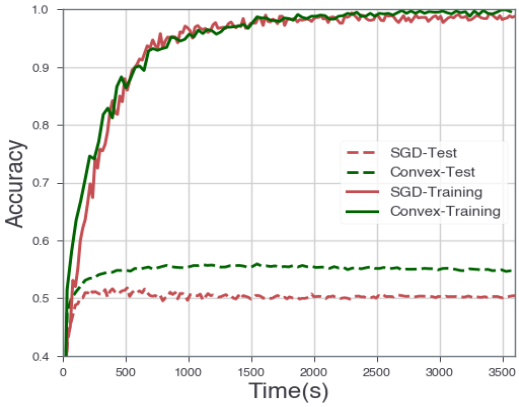
Remark 4. *For convolutional networks, we operate on the patch matrices $\{\mathbf{X}_b\}_{b=1}^B$ instead of $\mathbf{X}$, where $\mathbf{X}_b \in \mathbb{R}^{n \times h}$ and h denotes the filter size. Therefore, even when the data matrix is full rank, i.e., $r = d$, the number of relevant hyperplane arrangements in* (13) *is $\mathcal{O}(n^{r_c})$, where $r_c := rank([\mathbf{X}_1; \ldots; \mathbf{X}_B]) \leq h \ll d$. As an example, if we consider a convolutional network with m_1 3×3 filters, then $r_c \leq 9$ independent of the dimension d. As a corollary, this shows that the parameter sharing structure in CNNs significantly limits the number of hyperplane arrangements after ReLU activation, which might be one of the key factors behind their generalization performance in practice.*

Remark 5. *We can also compute the number of the hyperplane arrangements in the l^{th} layer, i.e., P_l. We first note that if we use the same approach for P_3 then due to the multiplicative structure in* (14)*, we have $P_3 \leq P_3'(2P_2)^{m_2} \leq \mathcal{O}(n^{m_2 m_1 r})$. Therefore, applying this relation recursively yields $P_l \leq \mathcal{O}(n^{r \prod_{j=1}^{l-1} m_j})$, which is also a polynomial term in both n and d for fixed data rank r and fixed width $\{m_j\}_{j=1}^{l-1}$.*

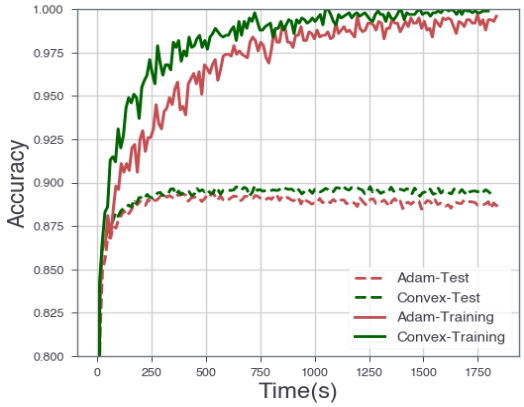
2.2. Training Complexity

In this section, we analyze the computational complexity to solve the convex problem in (12). We first note that (12) has $4dm_1P_1P_2$ variables and $4nP_1P_2(m_1+1)$ constraints. Thus, given the bound on the hyperplane arrangements in (13) and (14), a standard convex optimization solver, e.g., an interior point method, can globally optimize (12) with a polynomial-time complexity, i.e., $\mathcal{O}(d^3 m_1^3 n^{3(m_1+1)r})$. This result might also be extended to arbitrarily deep networks as detailed below.

Corollary 1. *Remark 5 shows that L-layer architectures can be globally optimized with $\mathcal{O}\left(d^3 \left(\prod_{j=1}^{L-2} m_j^3\right) n^{3r\left(1+\sum_{l=1}^{L-2} \prod_{j=1}^{l} m_j\right)}\right)$ complex-*



(a) CIFAR-10



(b) Fashion-MNIST

Figure 5: Accuracy of a three-layer architecture trained using the non-convex formulation (4) and the proposed convex program (12), where we use (a) CIFAR-10 with $(n, d, m_1, K, \beta, \text{batch size}) = (5\text{x}10^4, 3072, 100, 40, 10^{-3}, 10^3)$ and (b) Fashion-MNIST with $(n, d, m_1, K, \beta, \text{batch size}) = (6\text{x}10^4, 784, 100, 40, 10^{-3}, 10^3)$. We note that the convex model is trained using (a) SGD and (b) Adam and we use the approximation in Remark 3 for the convex programs.

ity, which is polynomial in n, d for fixed rank and widths.

3. Numerical Results

In this section[4], we present several numerical experiments validating our theory in the previous section. We first conduct an experiment on a synthetic dataset with $(n, d) = (5, 2)$. For this dataset, we first randomly generate d dimensional data samples $\{\mathbf{x}_i\}_{i=1}^n$ using a multivariate Gaussian distribution with zero mean and identity covariance, i.e., $\mathbf{x}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$. We then forward propagate these samples through a randomly initialized three-layer architecture with $m_1 = 3$ and $K = 5$ to obtain the corresponding labels $\mathbf{y} \in \mathbb{R}^n$. We then train the three-layer architecture in (4) on this synthetic dataset using SGD and our convex approach in (12). In Figure 4, we plot the training objective values with respect to the computation time taken by each algorithm, where we include 5 independent initialization trials for SGD. Moreover, for the convex approach, we plot both the objective value of the convex program in (12) and its non-convex equivalent constructed as described in Proposition 1. Here, we observe that when K is small, SGD trials tend to get stuck at a local minimum. Furthermore, as we increase the number of sub-networks, all the trials are able to converge to the global minimum achieved by the convex program. We also note that these observations are also consistent with the landscape visualizations in Figure 2.

In order to validate our theory, we also perform several experiments on some small scale real datasets available in UCI Machine Learning Repository (Dua & Graff, 2017). For these datasets, we consider a regression framework with $(m_1, K) = (7, 10)$ and compare the training and test performance of SGD and the convex program in (12). As reported in Table 1, our convex approach achieves a lower training objective for all the datasets. Although the convex approach also obtains lower test errors for five out of six datasets, there is a case where SGD achieves better test performance, i.e., the fertility dataset in Table 1. We believe that this is an interesting observation related to the generalization properties of SGD and the convex approach and leave this as an open problem for future research.

We also conduct an experiment on CIFAR-10 (Krizhevsky et al., 2014) and Fashion-MNIST (Xiao et al., 2017). Here, we consider a ten class classification problem using an architecture with $(m_1, K) = (100, 40)$, and report the test and training accuracies. In Figure 5, we provide these values with respect to time. This experiment verifies the performance boost provided by training on the convex formulations.

4. Concluding Remarks

We presented a convex analytic framework to characterize the optimal solutions to an architecture constructed by combining multiple deep ReLU networks in parallel. Particularly, we first derived an exact equivalent formulation for the non-convex primal problem using convex duality. This formulation has two significant advantages over the non-convex primal problem. First, since the equivalent problem is convex, it can be globally optimized by standard convex solvers without requiring any exhaustive search to tune hyperparameters, e.g., learning rate and initialization, or heuristics such as dropout. Second, we proved that glob-

[4] Additional experiments and details on the numerical results can be found in Section A.1 the supplementary file.

ally optimizing the equivalent problem has polynomial-time complexity with respect to the number of samples n and the feature dimension d. Therefore, we proved the polynomial-time trainability of regularized ReLU networks with more than two layers, which was previously known only for basic two-layer ReLU networks (Pilanci & Ergen, 2020). More importantly, since the equivalent problem is convex, one can achieve further interpretations and develop faster solvers by utilizing the tools in convex optimization.

Our approach also revealed an implicit regularization structure behind the original non-convex training problem. This structure is known as group ℓ_1-regularization that encourages sparsity between certain groups of parameters. As a corollary, the regularization in the convex problem implies that the original non-convex training problem achieves sparse solutions at global minima, where sparsity is over the number sub-networks. Our analysis also demystified mechanisms behind empirical observations regarding the convergence of SGD (see Figure 4) and loss landscape (see Figure 2 and (Zhang et al., 2019; Haeffele & Vidal, 2017)).

We conclude with the limitations of this work and some open research problems:

- The architectures studied in this work has three layers (two ReLU layers). Notice that even though we already provided some complexity results for deeper architectures, deriving the corresponding convex representations still remain an open problem.
- In order to utilize convex duality, we put unit ℓ_2-norm constraints on the first $L-2$ layer weights, which do not reflect the common practice. Therefore, we conjecture that weight decay regularization might not be the proper way of regularizing deep ReLU networks.
- When the data matrix is full rank, our approach has exponentia-time complexity, which is unavoidable unless $\mathrm{P}=\mathrm{NP}$ as detailed in (Pilanci & Ergen, 2020).

Acknowledgements

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

Anandkumar, A. and Ge, R. Efficient approaches for escaping higher order saddle points in non-convex optimization. In *Conference on learning theory*, pp. 81–102, 2016.

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

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

Bartlett, P. and Ben-David, S. Hardness results for neural network approximation problems. In *European Conference on Computational Learning Theory*, pp. 50–62. Springer, 1999.

Blum, A. and Rivest, R. L. Training a 3-node neural network is np-complete. In *Advances in neural information processing systems*, pp. 494–501, 1989.

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

Brutzkus, A. and Globerson, A. Globally optimal gradient descent for a convnet with gaussian inputs. *arXiv preprint arXiv:1702.07966*, 2017.

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

Chollet, F. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258, 2017.

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.

Czerniak, J. and Zarzycki, H. Application of rough sets in the presumptive diagnosis of urinary system diseases. In *Artificial intelligence and security in computing systems*, pp. 41–51. Springer, 2003.

DasGupta, B., Siegelmann, H. T., and Sontag, E. On the complexity of training neural networks with continuous activation functions. *IEEE Transactions on Neural Networks*, 6(6):1490–1504, 1995.

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

Dua, D. and Graff, C. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

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

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. 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 2020a. 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*, 2020b.

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

Ergen, T. and Pilanci, M. Convex duality of deep neural networks. *CoRR*, abs/2002.09773, 2020d. URL https://arxiv.org/abs/2002.09773.

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

Ergen, T., Sahiner, A., Ozturkler, B., Pauly, J. M., Mardani, M., and Pilanci, M. Demystifying batch normalization in relu networks: Equivalent convex optimization models and implicit regularization. *CoRR*, abs/2103.01499, 2021. URL https://arxiv.org/abs/2103.01499.

Ge, R., Lee, J. D., and Ma, T. Learning one-hidden-layer neural networks with landscape design, 2017.

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. *Deep learning*, volume 1. MIT press Cambridge, 2016.

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

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

Haeffele, B. D. and Vidal, R. Global optimality in neural network training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7331–7339, 2017.

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.

Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., and Keutzer, K. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

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.

Neyshabur, B., Li, Z., Bhojanapalli, S., LeCun, Y., and Srebro, N. Towards understanding the role of over-parametrization in generalization of neural networks. *arXiv preprint arXiv:1805.12076*, 2018.

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

Pilanci, M. and Ergen, T. Neural networks are convex regularizers: Exact polynomial-time convex optimization formulations for two-layer networks. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7695–7705. PMLR, 13–18 Jul 2020. URL http://proceedings.mlr.press/v119/pilanci20a.html.

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.

Safran, I. and Shamir, O. Spurious local minima are common in two-layer relu neural networks. In *International Conference on Machine Learning*, pp. 4433–4441. PMLR, 2018.

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

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.

Shalev-Shwartz, S., Shamir, O., and Shammah, S. Failures of gradient-based deep learning. *arXiv preprint arXiv:1703.07950*, 2017.

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.

Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. A. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

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.

Veit, A., Wilber, M. J., and Belongie, S. Residual networks behave like ensembles of relatively shallow networks. In *Advances in neural information processing systems*, pp. 550–558, 2016.

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

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500, 2017.

Zagoruyko, S. and Komodakis, N. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Zhang, H., Shao, J., and Salakhutdinov, R. Deep neural networks with multi-branch architectures are intrinsically less non-convex. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1099–1109, 2019.