Deep Neural Networks with Multi-Branch Architectures Are Intrinsically Less Non-Convex

Hongyang Zhang
Carnegie Mellon University
hongyanz@cs.cmu.edu

Junru Shao
Carnegie Mellon University
junrus@cs.cmu.edu

Ruslan Salakhutdinov
Carnegie Mellon University
rsalakhu@cs.cmu.edu

Abstract

Several recently proposed architectures of neural networks such as ResNeXt, Inception, Xception, SqueezeNet and Wide ResNet are based on the designing idea of having multiple branches and have demonstrated improved performance in many applications. We show that one cause for such success is due to the fact that the multi-branch architecture is intrinsically less non-convex in terms of duality gap. The duality gap measures the degree of intrinsic non-convexity of an optimization problem: smaller gap in relative value implies lower degree of intrinsic non-convexity. The challenge is to quantitatively measure the duality gap of highly non-convex problems such as deep neural networks. In this work, we provide strong guarantees of this quantity for two classes of network architectures. For the neural networks with arbitrary activation functions, multi-branch architecture and a variant of hinge loss, we show that the duality gap of both population and empirical risks shrinks to zero as the number of branches increases. This result sheds light on better understanding the power of over-parametrization where increasing the number of branches tends to make the loss surface less non-convex. For the neural networks with linear activation function and $\ell_2$ loss, we show that the duality gap of empirical risk is zero. Our two results work for arbitrary depths, while the analytical techniques might be of independent interest to non-convex optimization more broadly. Experiments on both synthetic and real-world datasets validate our results.

1 Introduction

Deep neural networks are a central object of study in machine learning, computer vision, and many other domains. They have substantially improved over conventional learning algorithms in many areas, including speech recognition, object detection, and natural language processing [28]. The focus of this work is to investigate the duality gap of deep neural networks. The duality gap is the discrepancy between the optimal values of primal and dual problems. While it has been well understood for convex optimization, little is known for non-convex problems. A smaller duality gap in relative value typically implies that the problem itself is less non-convex intrinsically, and thus is easier to optimize. Our results establish that: Deep neural networks with multi-branch architecture have small duality gap in relative value.

Our study is motivated by the computational difficulties of deep neural networks due to its non-convex nature. While many works have witnessed the power of local search algorithms for deep neural networks [16], these algorithms typically converge to a suboptimal solution in the worst cases according to various empirical observations [52, 28]. It is reported that for a single-hidden-layer neural network, when the number of hidden units is small, stochastic gradient descent may get easily stuck at the poor local minima [27, 49]. Furthermore, there is significant evidence indicating that when the networks are deep enough, bad saddle points do exist [1] and might be hard to escape [15, 21, 10, 1].

Given the computational obstacles, several efforts have been devoted to designing new architectures to alleviate the above issues, including over-parametrization [17, 53, 23, 41, 2, 45] and multi-branch architectures [64, 56, 18, 61, 33, 58]. Empirically, increasing the number of hidden units of a single-hidden-layer network

1 Throughout the paper, we discuss the duality gap w.r.t. the Lagrangian function, rather than the augmented Lagrangian function as in Chapter 11 of [18] where the duality gap is always zero.
encourages the first-order methods to converge to a global solution, which probably supports the folklore that the loss surface of a wider network looks more “convex” (see Figure 1). Furthermore, several recently proposed architectures, including ResNeXt [61], Inception [56], Xception [15], SqueezeNet [23], and Wide ResNet [62] are based on having multiple branches and have demonstrated substantial improvement over many of the existing models in many applications. In this work, we show that one cause for such success is due to the fact that the loss of multi-branch network is less non-convex in terms of duality gap.

Why is duality gap a measure of intrinsic non-convexity? Although some highly non-convex problems such as PCA and quadratic programming may have small/zero duality gap, we argue that the duality gap is a measure of intrinsic non-convexity of an optimization problem. There are two reasons for such an argument. a) The optimal value of the dual problem is equal to the optimal value of the convex relaxation of the primal problem. Hereby, the convex relaxation is the problem arising by replacing the non-convex objective with its convex closure and replacing the non-convex feasible set with its closed convex hull. Therefore, the duality gap measures the discrepancy between the optimal values of primal problem and its convex relaxation (Taking convex problems as an example, the duality gap is zero in most cases). When the duality gap is small, one can solve the convex relaxation problem whose solution is guaranteed to be close to the solution of primal problem. b) We show in our main result that the duality gap is a lower bound of the discrepancy between objective and its convex relaxation\footnote{Note that the convex relaxation of objective is different from the convex relaxation of primal non-convex problem which requires convexification operation on both objective and constraint.} (see Theorem 3 for the case of I = 1). So a smaller duality gap implies a possibly smaller discrepancy between objective and its convex relaxation.

Our Contributions. This paper provides both theoretical and experimental results for the population and empirical risks of deep neural networks by estimating the duality gap.

- We study the duality gap of deep neural networks with arbitrary activation functions, any data distribution, and multi-branch architecture (see Theorem 1). The multi-branch architecture is general, which includes the classic one-hidden-layer architecture as a special case (see Figure 2). By Shapley-Folkman lemma, we show that the duality gap of both population and empirical risks shrinks to zero as the number of branches increases.
- We prove that the strong duality (a.k.a. zero duality gap) holds for the empirical risk of deep linear neural networks (see Theorem 2). To this end, we develop multiple new proof techniques, including reduction to low-rank approximation and construction of dual certificate (see Section 3), which might be of independent interest to other non-convex problems.
- We empirically study the loss surface of multi-branch neural networks. Our experiments verify our theoretical findings.

Notation. We will use bold capital letter to represent matrix and lower-case letter to represent scalar. Specifically, let I be the identity matrix and denote by 0 the all-zero matrix. Let \( \{W_i \in \mathbb{R}^{d_i \times d_{i-1}} : i = 1, 2, ..., H\} \) be a set of network parameters, each of which represents the connection weights between the i-th and (i + 1)-th layers of neural network. We use \( W_{i,t} \in \mathbb{R}^{n_i \times 1} \) to indicate the t-th column of \( W \). We will use \( \sigma_i(W) \) to represent the i-th largest singular value of matrix \( W \). Given skinny SVD \( US\Sigma T \) of matrix \( W \), we denote by \( \text{svd}_r(W) = U_{r,1:r} \Sigma_{1:r,1:r} V_{1:r,1:r}^T \) the truncated SVD of \( W \) to the first r singular values. For matrix norms, denote by \( \|W\|_{S_H} = (\sum_{i} \sigma_i^H(W))^{1/H} \) the matrix Schatten-H norm. Nuclear norm and Frobenius norm are special cases of Schatten-H norm: \( \|W\|_1 = \|W\|_{S_1} \) and \( \|W\|_F = \|W\|_{S_2} \). We use \( \|W\| \) to represent the
matrix operator norm, i.e., $\|W\| = \sigma_1(W)$, and denote by $\text{rank}(W)$ the rank of matrix $W$. Denote by $\text{Row}(W)$ the span of rows of $W$. Let $W^\dagger$ be the Moore-Penrose pseudo-inverse of $W$.

For convex matrix function $K(\cdot)$, we denote by $K^*(A) = \max_M \langle A, M \rangle - K(M)$ the conjugate function of $K(\cdot)$ and $\partial K(\cdot)$ the sub-differential. We use $\text{diag}(\sigma_1, ..., \sigma_r)$ to represent a $r \times r$ diagonal matrix with diagonal entries $\sigma_1, ..., \sigma_r$. Let $d_{\min} = \min\{d_i : i = 1, 2, ..., H - 1\}$, and $[I] = \{1, 2, ..., I\}$. For any two matrices $A$ and $B$ of matching dimensions, we denote by $[A; B]$ the concatenation of $A$ and $B$ along the row and $[A; B]$ the concatenation of two matrices along the column.

2 Duality Gap of Multi-Branch Neural Networks

We first study the duality gap of neural networks in a classification setting. We show that the wider the network is, the smaller the duality gap becomes.

Network Setup. The output of our network follows from a multi-branch architecture (see Figure 2):

$$f(w; x) = \frac{1}{I} \sum_{i=1}^{I} f_i(w_{(i)}; x), \ w_{(i)} \in W_i,$$

where $W_i$ is a convex set, $w$ is the concatenation of all network parameters $\{w_{(i)}\}_{i=1}^{I}, x \in \mathbb{R}^{d_0}$ is the input instance, $\{W_i\}_{i=1}^{I}$ is the parameter space, and $f_i(w_{(i)}; \cdot)$ represents an $\mathbb{R}^{d_0} \to \mathbb{R}$ continuous mapping by a sub-network which is allowed to have arbitrary architecture such as convolutional and recurrent neural networks. As an example, $f_i(w_{(i)}; \cdot)$ can be in the form of a $H_i$-layer feed-forward sub-network:

$$f_i(w_{(i)}; x) = w_{(i)}^\top \psi_{H_i} (W_{(i)}^1 \psi_{H_i} (W_{(i)}^2 \psi_{H_i} (W_{(i)}^H x) \in \mathbb{R},$$

$$w_{(i)} = [w_i; \text{vec}(W_{(i)}^1); ..., \text{vec}(W_{(i)}^H)] \in \mathbb{R}^{p_i}.$$  

Hereby, the functions $\psi_k(\cdot), k = 1, 2, ..., H_i$ are allowed to encode arbitrary form of continuous element-wise non-linearity (and linearity) after each matrix multiplication, such as sigmoid, rectification, convolution, while the number of layers $H_i$ in each sub-network can be arbitrary as well. When $H_i = 1$ and $d_{H_i} = 1$, i.e., each sub-network in Figure 2 represents one hidden unit, the architecture $f(w; x)$ reduces to a one-hidden-layer network. We apply the so-called $\tau$-hinge loss [4, 7] on the top of network output for label $y \in \{-1, +1\}$:

$$\ell_\tau(w; x, y) := \max \left(0, 1 - \frac{y \cdot f(w; x)}{\tau}\right), \ \tau > 0.$$

The $\tau$-hinge loss has been widely applied in active learning of classifiers and margin based learning [4, 7].

When $\tau = 1$, it reduces to the classic hinge loss [33, 38].

We make the following assumption on the margin parameter $\tau$, which states that the parameter $\tau$ is sufficiently large.

Assumption 1 (Parameter $\tau$). For sample $(x, y)$ drawn from distribution $P$, we have $\tau > y \cdot f(w; x)$ for all $w \in W_1 \times W_2 \times ... \times W_I$ with probability measure 1.

We further empirically observe that using smaller values of the parameter $\tau$ and other loss functions support our theoretical result as well (see experiments in Section 5). It is an interesting open question to extend our theory to more general losses in the future.

To study how close these generic neural network architectures approach the family of convex functions, we analyze the duality gap of minimizing the risk w.r.t. the loss [1] with an extra regularization constraint. The normalized duality gap is a measure of intrinsic non-convexity of a given function [13]: the gap is zero when the given function itself is convex, and is large when the loss surface is far from the convexity intrinsically. Typically, the closer the network approaches to the family of convex functions, the easier we can optimize the network.

Multi-Branch Architecture. Our analysis of multi-branch neural networks is built upon tools from non-convex geometric analysis — Shapley–Folkman lemma. Basically, the Shapley–Folkman lemma states that the sum of constrained non-convex functions is close to being convex. A neural network is an ideal target to apply this lemma to: the width of network is associated with the number of summand functions. So intuitively, the wider the neural network is, the smaller the duality gap will be. In particular, we study the following non-
Deep Neural Networks with Multi-Branch Architectures Are Intrinsically Less Non-Convex

convex problem concerning the population risk:

\[
\min_{w \in W_1 \times \cdots \times W_I} \mathbb{E}_{(x,y) \sim P} [f_r(w; x, y)],
\]
\[\text{s.t.} \quad \frac{1}{I} \sum_{i=1}^{I} h_i(w(i)) \leq K, \tag{2}\]

where \(h_i(\cdot), i \in [I]\) are convex regularization functions, e.g., the weight decay, and \(K\) can be arbitrary such that the problem is feasible. Correspondingly, the dual problem of problem (2) is a one-dimensional convex optimization problem:

\[
\max_{\lambda \geq 0} Q(\lambda) - \lambda K,
\]

\[
Q(\lambda) := \inf_{w \in W_1 \times \cdots \times W_I} \mathbb{E}_{(x,y) \sim P} [f_r(w; x, y)] + \lambda \frac{1}{I} \sum_{i=1}^{I} h_i(w(i)). \tag{3}\]

Before proceeding, we first define some notations to be used in our main results. For \(w \in W_i\), denote by

\[
\tilde{f}_i(w) := \inf_{a^1, w^j(i) \in W_i} \sum_{j=1}^{p_i+2} a^j \mathbb{E}_{(x,y) \sim P} \left( 1 - \frac{y \cdot f_i(w^j(i); x)}{\tau} \right)
\]
\[\text{s.t.} \quad w = \sum_{j=1}^{p_i+2} a^j w^j(i), \quad a^j \geq 0.\]

This represents the convex relaxation of the \(i\)-th summand term \(\mathbb{E}_{(x,y) \sim P} [1 - y \cdot f_i(\cdot; x)/\tau]\) in the objective, because the epigraph of \(\tilde{f}_i\) is exactly the convex hull of epigraph of \(\mathbb{E}_{(x,y) \sim P} [1 - y \cdot f_i(\cdot; x)/\tau]\) by the definition of \(\hat{f}_i\). For \(\tilde{w} \in W_i\), we also define

\[
\hat{f}_i(\tilde{w}) := \inf_{w^j(i) \in W_i} \mathbb{E}_{(x,y) \sim P} \left( 1 - \frac{y \cdot f_i(w^j(i); x)}{\tau} \right)
\]
\[\text{s.t.} \quad h_i(w^j(i)) \leq h_i(\tilde{w}).\]

This is a “restricted” version of the \(i\)-th summand term \(\mathbb{E}_{(x,y) \sim P} [1 - y \cdot f_i(\cdot; x)/\tau]\) to the hard constraint \(h_i(w^j(i)) \leq h_i(\tilde{w})\).

Our main results for multi-branch neural networks are as follows:

**Theorem 1.** Denote by \(\inf(P)\) the minimum of primal problem (2) and \(\sup(D)\) the maximum of dual problem (3). Let \(\Delta_i := \sup_{w \in W_i} \left\{ \tilde{f}_i(w) - \hat{f}_i(w) \right\} \geq 0\) and \(\Delta_{\text{worst}} := \max_{i \in [I]} \Delta_i\). Suppose \(W_i\)'s are compact and both \(\tilde{f}_i(w^j(i); x)\) and \(h_i(w^j(i))\) are continuous w.r.t. \(w^j(i)\). If there exists at least one feasible solution of problem (P), then under Assumption (2) the duality gap w.r.t. problems (2) and (3) can be bounded by

\[
0 \leq \frac{\inf(P) - \sup(D)}{\Delta_{\text{worst}}} \leq \frac{2}{I}.
\]

**Remark 1.** Note that \(\Delta_i\) measures the divergence between the function value of \(\tilde{f}_i\) and its convex relaxation \(f_i\). The constant \(\Delta_{\text{worst}}\) is the maximal divergence among all sub-networks, which grows slowly with the increase of \(I\). This is because \(\Delta_{\text{worst}}\) only measures the divergence of one branch. The normalized duality gap \((\inf(P) - \sup(D))/\Delta_{\text{worst}}\) has been widely used before to measure the degree of non-convexity of optimization problems \([13, 57, 12, 24, 22]\). Such a normalization avoids trivialities in characterizing the degree of non-convexity: scaling the objective function by any constant does not change the value of normalized duality gap.

**Remark 2.** Even though Theorem (7) is in the form of population risk, the conclusion still holds for the empirical loss as well. This can be achieved by setting the marginal distribution \(P_x\) as the uniform distribution on a finite set and \(P_y\) as the corresponding labels uniformly distributed on the same finite set.

**Remark 3.** Setting \(K\) in problem (2) infinitely large implies that Theorem (7) holds for unconstrained deep neural networks as well.

**Inspiration for Architecture Designs.** Theorem (7) shows that the duality gap of deep network shrinks when the width \(I\) is large: when \(I \to +\infty\), surprisingly, deep network is as easy as a convex optimization, as the gap is zero. An intuitive explanation is that the large number of randomly initialized hidden units represent all possible features. Thus the optimization problem involves just training the top layer of the network, which is convex. Our result encourages a class of network architectures with multiple branches and supports some of the most successful architectures in practice, such as Inception \([56]\), Xception \([18]\), ResNeXt \([61]\), SqueezeNet \([33]\), Wide ResNet \([62]\), Shake-Shake regularization \([25]\) — all of which benefit from the split-transform-merge behaviour as shown in Figure 2.

**Related Works.** While many efforts have been devoted to studying the local minima or saddle points of deep neural networks \([42, 68, 54, 56, 60, 69]\), little is known about the duality gap of deep networks. In particular, Choromanska et al. \([20, 19]\) showed that the number of poor local minima cannot be too large. Kawaguchi \([35]\) improved over the results of \([20, 19]\) by assuming that the activation functions are independent Bernoulli variables and the input data are drawn from Gaussian distribution. Xie et al. \([60]\) and Haefele et al. \([30]\) studied the local minima of regularized network, but they require either the network is shallow, or
the network weights are rank-deficient. Ge et al. [27] showed that every local minimum is globally optimal by modifying the activation function. Zhang et al. [67] and Aslan et al. [3] reduced the non-linear activation to the linear case by kernelization and relaxed the non-convex problem to a convex one. However, no formal guarantee was provided for the tightness of the relaxation. Theorem 1 on the other hand, bounds the duality gap of deep neural networks with mild assumptions.

Another line of research studies the convexity behaviour of neural networks when the number of hidden neurons goes to the infinity. In particular, Bach [5] proved that a single-hidden-layer network is as easy as a convex optimization by using classical non-Euclidean regularization tools. Bengio et al. [12] showed a similar phenomenon for multi-layer networks with an incremental algorithm. In comparison, Theorem 1 not only captures the convexification phenomenon when $I \to +\infty$, but also goes beyond the result as it characterizes the convergence rate of convexity of neural networks in terms of duality gap. Furthermore, the conclusion in Theorem 1 holds for the population risk, which was unknown before.

3 Strong Duality of Linear Neural Networks

In this section, we show that the duality gap is zero if the activation function is linear. Deep linear neural networks have received significant attention in recent years [31, 35, 67, 44, 8, 28, 31, 9] because of its simple formulation and its connection to non-linear neural networks.

Network Setup. We discuss the strong duality of regularized linear neural networks of the form

$$\begin{align*}
(W_1^*, \ldots, W_H^*) &= \arg\min_{W_1, \ldots, W_H} \frac{1}{2} \|Y - W_H \cdots W_1 X\|_F^2 \\
&+ \frac{\gamma}{H} \left( \|W_1 X\|_{S_H} + \sum_{i=2}^H \|W_i\|_{S_H} \right),
\end{align*}$$

where $X = [x_1, \ldots, x_n] \in \mathbb{R}^{d_0 \times n}$ is the given instance matrix, $Y = [y_1, \ldots, y_n] \in \mathbb{R}^{d_H \times n}$ is the given label matrix, and $W_i \in \mathbb{R}^{d_i \times d_{i-1}}$, $i \in [H]$ represents the weight matrix in each linear layer. We mention that (a) while the linear operation is simple matrix multiplications in problem (4), it can be easily extended to other linear operators, e.g., the convolutional operator or the linear operator with the bias term, by properly involving a group of kernels in the variable $W_i$ [30]. (b) The regularization terms in problem (4) are of common interest, e.g., see [30]. When $H = 2$, our regularization terms reduce to $\frac{1}{2} \|W_i\|_F^2$, which is well known as the weight-decay or Tikhonov regularization. (c) The regularization parameter $\gamma$ is the same for each layer since we have no further information on the preference of layers.

Our analysis leads to the following guarantees for the deep linear neural networks.

Theorem 2. Denote by $\tilde{Y} := YY^T \in \mathbb{R}^{d_H \times n}$ and $d_{\min} := \min\{d_1, \ldots, d_{H-1}\} \leq \min\{d_0, d_H, n\}$. Let $0 \leq \gamma < \sigma_{\min}(\tilde{Y})$ and $H \geq 2$, where $\sigma_{\min}(\tilde{Y})$ stands for the minimal non-zero singular value of $\tilde{Y}$. Then the strong duality holds for deep linear neural networks (4). In other words, the optimum of problem (4) is the same as its convex dual problem

$$\begin{align*}
\Lambda^* = \arg\max_{\Lambda \subseteq \text{Row}(X)} -\frac{1}{2} \|\tilde{Y} - \Lambda\|_{d_{\min}}^2 + \frac{1}{2} \|\tilde{Y}\|_F^2, \\
s.t. \|\Lambda\| \leq \gamma,
\end{align*}$$

where $\|\cdot\|_{d_{\min}} = \sum_{i=1}^{d_{\min}} \sigma_i^2(\cdot)$ is a convex function. Moreover, the optimal solutions of primal problem (4) can be obtained from the dual problem (5) in the following way: let $U \Sigma V^T = \text{svd}_{d_{\min}}(\tilde{Y} - \Lambda^*)$ be the skinny SVD of matrix $\text{svd}_{d_{\min}}(\tilde{Y} - \Lambda^*)$, then $W_i^* = [\Sigma_i^{1/H}, \mathbf{0}; \mathbf{0}] \in \mathbb{R}^{d_i \times d_{H-1}}$ for $i = 2, 3, \ldots, H-1$, $W_H^* = [U \Sigma_i^{1/H}, \mathbf{0}] \in \mathbb{R}^{d_H \times d_{H-2}}$ and $W_i^* = [\Sigma_i^{1/H} V_T; \mathbf{0}] \in \mathbb{R}^{d_i \times d_0}$ is a globally optimal solution to problem (4).

The regularization parameter $\gamma$ cannot be too large in order to avoid underfitting. Our result provides a suggested upper bound $\sigma_{\min}(\tilde{Y})$ for the regularization parameter, where oftentimes $\sigma_{\min}(\tilde{Y})$ characterizes the level of random noise. When $\gamma = 0$, our analysis reduces to the un-regularized deep linear neural network, a model which has been widely studied in [33, 44, 8, 28].

Theorem 2 implies the following result on the landscape of deep linear neural networks: the regularized deep learning can be converted into an equivalent convex problem by dual. To the best of our knowledge, this is the first result on the strong duality of linear neural networks. We note that the strong duality rarely happens in the non-convex optimization: matrix completion [3], Fantope [47], and quadratic optimization with two quadratic constraints [11] are among the few paradigms that enjoy the strong duality. For deep networks, the effectiveness of convex relaxation has been observed empirically in [3, 67], but much remains unknown for the theoretical guarantees of the relaxation. Our work shows strong duality of regularized deep linear neural networks and provides an alternative approach to overcome the computational obstacles due
Deep Neural Networks with Multi-Branch Architectures Are Intrinsically Less Non-Convex

Figure 3: Visualization of Shapley-Folkman lemma. **The first figure:** an $\ell_{1/2}$ ball. **The second and third figures:** the averaged Minkowski sum of two and ten $\ell_{1/2}$ balls. **The fourth figure:** the convex hull of $\ell_{1/2}$ ball (the Minkowski average of infinitely many $\ell_{1/2}$ balls). It show that with the number of $\ell_{1/2}$ balls to be averaged increasing, the Minkowski average tends to be more convex.

to the non-convexity: one can apply convex solvers, e.g., the Douglas–Rachford algorithm\(^1\) for problem (5) and then conduct singular value decomposition to compute the weights $\{W^*_i\}_{i=1}^H$ from $\text{svd}_{\text{min}}(Y - \Lambda^*)$. In addition, our result inherits the benefits of convex analysis. The vast majority results on deep learning study the generalization error or expressive power by analyzing its complicated non-convex form \([46, 63]\). In contrast, with strong duality one can investigate various properties of deep linear networks with much simpler convex form.

**Related Works.** The goal of convexified linear neural networks is to relax the non-convex form of deep learning to the computable convex formulations \([67, 3]\). While several efforts have been devoted to investigating the effectiveness of such convex surrogates, e.g., by analyzing the generalization error after the relaxation \([67]\), little is known whether the relaxation is tight to its original problem. Our result, on the other hand, provides theoretical guarantees for the tightness of convex relaxation of deep linear networks, a phenomenon observed empirically in \([3, 67]\).

We mention another related line of research — no bad local minima. On one hand, although recent works have shown the absence of spurious local minimum for deep linear neural networks \([53, 35, 11]\), many of them typically lack theoretical analysis of regularization term. Specifically, Kawaguchi \([35]\) showed that un-regularized deep linear neural networks have no spurious local minimum. Lu and Kawaguchi \([11]\) proved that depth creates no bad local minimum for un-regularized deep linear neural networks. In contrast, our optimization problem is more general by taking the regularization term into account. On the other hand, even the “local–global” argument holds for the deep linear neural networks, it is still hard to escape bad saddle points \([1]\).

In particular, Kawaguchi \([35]\) proved that for linear networks deeper than three layers, there exist bad saddle points at which the Hessian does not have any negative eigenvalue. Therefore, the state-of-the-art algorithms designed to escape the saddle points might not be applicable \([33, 26]\). Our result provides an alternative approach to solve deep linear network by convex programming.

4 Our Techniques and Proof Sketches

In this section, we present our techniques and proof sketches of Theorems 1 and 2.

(a) **Shapley-Folkman Lemma.** The proof of Theorem 1 is built upon the Shapley-Folkman lemma \([22, 55, 24, 13]\), which characterizes a convexification phenomenon concerning the average of multiple sets and is analogous to the central limit theorem in the probability theory. Consider the averaged Minkowski sum of $I$ sets $A_1, A_2, ..., A_I$ given by $\{I^{-1} \sum_{j \in [I]} a_j : a_j \in A_j\}$. Intuitively, the lemma states that $\rho(I^{-1} \sum_{j \in [I]} A_j) \to 0$ as $I \to +\infty$, where $\rho(\cdot)$ is a metric of the non-convexity of a set (see Figure 3 for visualization). We apply this lemma to the optimization formulation of deep neural networks. Denote by augmented epigraph the set $\{(h(w), \ell(w)) : \text{all possible choices of } w\}$, where $h$ is the constraint and $\ell$ is the objective function in the optimization problem. The key observation is that the augmented epigraph of neural network loss with multi-branch architecture can be expressed as the Minkowski average of augmented epigraphs of all branches. Thus we obtain a natural connection between an optimization problem and its corresponding augmented epigraph. Applying Shapley-Folkman lemma to the augmented epigraph leads to a characteristic of non-convexity of the deep neural network.

(b) **Variational Form.** The proof of Theorem 2 is built upon techniques (b), (c), and (d). In particular, problem (4) is highly non-convex due to its multi-linear form over the optimized variables $\{W_i\}_{i=1}^H$. Fortu-
nately, we are able to analyze the problem by grouping \( W_H W_{H-1} \cdots W_1 \) together and converting the original non-convex problem in terms of the separate variables \( \{ W_{[i]} \}_{i=1}^H \) to a convex optimization with respect to the new grouping variable \( W_H W_{H-1} \cdots W_1 \). This typically requires us to represent the objective function of (4) as a convex function of \( W_H W_{H-1} \cdots W_1 \). To this end, we prove that \( \| W_H W_{H-1} \cdots W_1 X \|_F^2 \) is minimized when

\[
\min_{W_1, \ldots, W_H} \frac{1}{2} \| Y - W_H W_{H-1} \cdots W_1 X \|_F^2
\]

for a fixed Lagrangian variable \( \Lambda^* \), which we will construct later in subsection (d). While

\[
\min_{W_1, \ldots, W_H} \frac{1}{2} \| Y - \Lambda^* - W_H W_{H-1} \cdots W_1 X \|_F^2
\]

for any \( \Lambda^* \), is the number of hidden neurons, and \( w \) is the number of hidden neurons in the network, no non-convexity can be observed on the extreme case, when there are \( I \) hidden neurons [49]. We make use of the visualization technique employed by [40] to plot the landscape, where we project the high-dimensional hinge loss \((\tau = 1)\) landscape onto a 2-d plane spanned by three points. These points are found by running the SGD algorithm with three different initializations until the algorithm converges. As shown in Figure 4, the landscape exhibits strong non-convexity with lots of local minima in the under-parameterized case \( I = 10 \). But as \( I \) increases, the landscape becomes more convex. In the extreme case, when there are 1,000 hidden neurons in the network, no non-convexity can be observed on the landscape.

The desired strong duality is a straightforward result from this argument.

(d) Dual Certificate. The remaining proof is to construct a dual certificate \( \Lambda^* \) such that the dual conditions hold true. The challenge is that the dual conditions impose several constraints simultaneously on the dual certificate, making it hard to find a desired certificate. This is why progress on the dual certificate has focused on convex programming. To resolve the issue, we carefully choose the certificate as an appropriate scaling of subgradient of nuclear norm around a low-rank solution, where the nuclear norm follows from our regularization term in technique (b). Although the nuclear norm has infinitely many subgradients, we prove that our construction of dual certificate obeys all desired dual conditions. Putting techniques (b), (c), and (d) together, our proof of strong duality is completed.

5 Experiments

In this section, we verify our theoretical contributions by the experimental validation.

5.1 Visualization of Loss Landscape

Experiments on Synthetic Datasets. We first show that over-parametrization results in a less non-convex loss surface for a synthetic dataset. The dataset consists of 1,000 examples in \( \mathbb{R}^{10} \) whose labels are generated by an underlying one-hidden-layer ReLU network \( f(x) = \sum_{i=1}^I w_{i,2}^* | W_{i,1}^* x_+ \) with 11 hidden neurons [49]. We make use of the visualization technique employed by [40] to plot the landscape, where we project the high-dimensional hinge loss \((\tau = 1)\) landscape onto a 2-d plane spanned by three points. These points are found by running the SGD algorithm with three different initializations until the algorithm converges. As shown in Figure 4, the landscape exhibits strong non-convexity with lots of local minima in the under-parameterized case \( I = 10 \). But as \( I \) increases, the landscape becomes more convex. In the extreme case, when there are 1,000 hidden neurons in the network, no non-convexity can be observed on the landscape.

Experiments on MNIST and CIFAR-10. We next verify the phenomenon of over-parametrization on MNIST [49] and CIFAR-10 [27] datasets. For both datasets, we follow the standard preprocessing step that each pixel is normalized by subtracting its mean and dividing by its standard deviation. We do not apply data augmentation. For MNIST, we consider a single-hidden-layer network defined as: \( f(x) = \sum_{i=1}^I w_{i,2} W_{i,1} x_+ \), where \( W_{i,1} \in \mathbb{R}^{h \times d} \) and \( W_{i,2} \in \mathbb{R}^{10 \times h} \), \( d \) is the input dimension, \( h \) is the number of hidden neurons, and \( I \) is...
This observation enables us to represent the optimization problem as a convex function of the output of a neural network. Therefore, we can analyze the non-convexity of loss surfaces, which do not apply to the non-smooth problem. In this work, we study the duality gap for two classes of network architectures. For the neural network with arbitrary activation functions, multi-branch architecture and τ-hinge loss, we show that the duality gap of both population and empirical risks shrinks to zero as the number of branches increases. For the neural network with linear activation function and ℓ₂ loss, we show that the duality gap is zero. Our two results work for arbitrary depths, while the analytical techniques might be of independent interest to non-convex optimization more broadly, e.g., the optimization problems in adversarial defense and attack.

### Table 1: Frequency of hitting global minimum by SGD with 100 different initialization seeds.

<table>
<thead>
<tr>
<th># Hidden Neurons</th>
<th>Hitting Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>21 / 100</td>
</tr>
<tr>
<td>13</td>
<td>24 / 100</td>
</tr>
<tr>
<td>14</td>
<td>24 / 100</td>
</tr>
<tr>
<td>15</td>
<td>29 / 100</td>
</tr>
<tr>
<td>16</td>
<td>30 / 100</td>
</tr>
<tr>
<td>17</td>
<td>32 / 100</td>
</tr>
<tr>
<td>18</td>
<td>35 / 100</td>
</tr>
<tr>
<td>19</td>
<td>52 / 100</td>
</tr>
<tr>
<td>20</td>
<td>64 / 100</td>
</tr>
<tr>
<td>21</td>
<td>75 / 100</td>
</tr>
</tbody>
</table>

### 5.2 Frequency of Hitting Global Minimum

To further analyze the non-convexity of loss surfaces, we consider various one-hidden-layer networks, where each network was trained 100 times using different initialization seeds under the setting discussed in our synthetic experiments of Section 5.1. Since we have the ground-truth global minimum, we record the frequency that SGD hits the global minimum up to a small error $1 \times 10^{-5}$ after 100,000 iterations. Table 1 shows that increasing the number of hidden neurons results in higher hitting rate of global optimality. This further verifies that the loss surface of one-hidden-layer neural network becomes less non-convex as the width increases.
References


Deep Neural Networks with Multi-Branch Architectures Are Intrinsically Less Non-Convex


