On the Power of Over-parametrization in Neural Networks with Quadratic Activation

Simon S. Du [1]   **Jason D. Lee** [2]

Abstract

We provide new theoretical insights on why over-parametrization is effective in learning neural networks. For a k hidden node shallow network with quadratic activation and n training data points, we show as long as $k \geq \sqrt{2n}$, over-parametrization enables local search algorithms to find a *globally* optimal solution for general smooth and convex loss functions. Further, despite that the number of parameters may exceed the sample size, using theory of Rademacher complexity, we show with weight decay, the solution also generalizes well if the data is sampled from a regular distribution such as Gaussian. To prove when $k \geq \sqrt{2n}$, the loss function has benign landscape properties, we adopt an idea from smoothed analysis, which may have other applications in studying loss surfaces of neural networks.

1. Introduction

Neural networks have achieved a remarkable impact on many applications such computer vision, reinforcement learning and natural language processing. Though neural networks are successful in practice, their theoretical properties are not yet well understood. Specifically, there are two intriguing empirical observations that existing theories cannot explain.

- **Optimization**: Despite the highly non-convex nature of the objective function, simple first-order algorithms like stochastic gradient descent are able to minimize the training loss of neural networks. Researchers have conjectured that the use of over-parametrization (Livni et al., 2014; Safran and Shamir, 2017) is the primary

[1]Machine Learning Department, Carnegie Mellon University [2]Department of Data Sciences and Operations, University of Southern California. Correspondence to: Simon S. Du <ssdu@cs.cmu.edu>.

reason why local search algorithms can achieve low training error. The intuition is over-parametrization alters the loss function to have a large manifold of globally optimal solutions, which in turn allows local search algorithms to more easily find a global optimal solution.

- **Generalization**: From the statistical point of view, over-parametrization may hinder effective generalization, since it greatly increases the number of parameters to the point of having number of parameters exceed the sample size. To address this, practitioners often use explicit forms of regularization such as weight decay, dropout, or early stopping to improve generalization. However in the non-convex setting, theoretically, we do not have a good quantitative understanding on how these regularizations help generalization for neural network models.

In this paper, we provide new theoretical insights into the optimization landscape and generalization ability of over-parametrized neural networks. Specifically we consider the neural network of the following form:

$$f(\mathbf{x}, \mathbf{W}) = \sum_{j=1}^{k} a_i \sigma\left(\langle \mathbf{w}_j, \mathbf{x} \rangle\right). \tag{1}$$

In the above $\mathbf{x} \in \mathbb{R}^d$ is the input vector, $\mathbf{W} \in \mathbb{R}^{d \times k}$ with $\mathbf{w}_j \in \mathbb{R}^d$ denotes the j-th row of $\mathbf{W}$ and a_i's are the weights in the second layer. Finally $\sigma(\cdot) : \mathbb{R} \to \mathbb{R}$ denotes the activation function applied to each hidden node. When the neural network is over-parameterized, the number of hidden notes k can be very large compared with input dimension d or the number of training samples.

In our setting, we fix the second layer to be $\mathbf{a} = (1, \ldots, 1)$. Although it is simpler than the case where the second layer is not fixed, the effect of over-parameterization can be studied in this setting as well because we do not have any restriction on the number of hidden nodes.

We focus on quadratic activation function $\sigma(z) = z^2$. Though quadratic activations are rarely used in practice, stacking multiple such two-layer blocks can be used to simulate higher-order polynomial neural networks and sigmodial

activated neural networks (Livni et al., 2014; Soltani and Hegde, 2017).

In practice, we have n training samples $\{\mathbf{x}_i, y_i\}_{i=1}^n$ and solve the following optimization problem to learn a neural network

$$\min_{\mathbf{W}} \frac{1}{n} \sum_{i=1}^n \ell\left(f(\mathbf{x}_i, \mathbf{W}), y_i\right)$$

where $\ell(\cdot, \cdot)$ is some loss function such as ℓ_2 or logistic loss. For gradient descent we use the following update

$$\mathbf{W} \leftarrow \mathbf{W} - \eta \sum_{i=1}^n \nabla_{\mathbf{W}} \ell\left(f(x_i), y_i\right)$$

where η is the step size.

To improve the generalization ability, we often add explicit regularization. In this paper, we focus on a particular regularization technique, weight decay for which we slightly change the gradient descent algorithm to

$$\mathbf{W} \leftarrow \mathbf{W} - \eta \sum_{i=1}^n \nabla_{\mathbf{W}} \ell\left(f(\mathbf{x}_i, \mathbf{W}), y_i\right) - \eta \lambda \mathbf{W}.$$

where λ is the decay rate. Note this algorithm is equivalent to applying the gradient descent algorithm on the regularized loss

$$\min_{\mathbf{W}} L\left(\mathbf{W}\right) = \frac{1}{n} \sum_{i=1}^n \ell\left(f(\mathbf{x}_i), y_i\right) + \frac{\lambda}{2} \|\mathbf{W}\|_F^2. \quad (2)$$

In this setup, we make the following theoretical contributions to explain why over-parametrization helps optimization and still allows for generalization.

1.1. Main Contributions

Over-parametrization Helps Optimization. We analyze two kinds of over-parameterization. First we show that for

$$k \geq d,$$

then all local minima in Problem (2) is global and all saddle points are strict. This properties together with recent algorithmic advances in non-convex optimization (Lee et al., 2016) imply gradient descent can find a globally optimal solution with random initialization. This is a minor generalization of results in (Soltanolkotabi et al., 2017) which only includes ℓ_2 loss, and (Haeffele and Vidal, 2015; Haeffele et al., 2014) which only include $k \geq d + 1$.

Second, we consider another form of over-parametrization,

$$\frac{k(k+1)}{2} > n.$$

This condition on the amount of over-parameterization is much milder than $k \geq n$, a condition used in many previous papers (Nguyen and Hein, 2017a;b). Further in practice, $k(k+1)/2 > n$ is a much milder requirement than $k \geq d$, since if $k \approx \sqrt{2n}$ and $n << d^2$ then $k << d$. In this setting, we consider the *perturbed* version of the Problem (2):

$$\min_{\mathbf{W}} L_{\mathbf{C}}\left(\mathbf{W}\right) = \frac{1}{n} \sum_{i=1}^n \ell\left(f(\mathbf{x}_i), y_i\right) + \frac{\lambda}{2} \|\mathbf{W}\|_F^2 \\ + \langle \mathbf{C}, \mathbf{W}^\top \mathbf{W} \rangle \quad (3)$$

where $\mathbf{C}$ is a random positive semidefinite matrix with *arbitrarily* small Frobenius norm. We show that if $\frac{k(k+1)}{2} > n$, Problem (3) also has the desired properties that all local minima are global and all saddle points are strict with probability 1. Since $\mathbf{C}$ has small Frobenius norm, the optimal value of Problem (3) is very close to that of Problem (2). See Section 3 for the precise statement.

To prove this surprising fact, we bring forward ideas from smoothed analysis in constructing the perturbed loss function (3), which we believe is useful for analyzing the landscape of non-convex losses.

Weight-decay Helps Generalization. We show because of weight-decay, the optimal solution of Problem (2) also generalizes well. The major observation is weight-decay ensures the solution of Problem (2) has low *Frobenius* norm, which is equivalent to matrix $\mathbf{W}^\top \mathbf{W}$ having low *nuclear* norm (Srebro et al., 2005). This observation allows us to use theory of Rademacher complexity to directly obtain quantitative generalization bounds. Our theory applies to a wide range of data distribution and in particular, does not need to assume the model is realizable. Further, the generalization bound does not depend on the number of epochs SGD runs or the number of hidden nodes.

To sum up, in this paper we justify the following folklore.

Over-parametrization allows us to find global optima and with weight decay, the solution also generalizes well.

1.2. Organization

This paper is organized as follows. In Section 2 we introduce necessary background and definitions. In Section 3 we present our main theorems on why over-parametrization helps optimization when $k \geq d$ *or* $\frac{k(k+1)}{2} > n$. In Section 4, we give quantitative generalization bounds to explain why weight decay helps generalization in the presence of over-parametrization. In Section 5, we prove our main theorems. We conclude and list future works in Section 6.

1.3. Related Works

Neural networks have enjoyed great success in many practical applications (Krizhevsky et al., 2012; Dauphin et al., 2016; Silver et al., 2016). To explain this success, many works have studied the expressiveness of neural networks. The expressive ability of shallow neural network dates back to 90s (Barron, 1994). Recent results give more refined analysis on deeper models (Bölcskei et al., 2017; Telgarsky, 2016; Wiatowski et al., 2017).

However, from the point of view of learning theory, it is well known that training a neural network is hard in the worst case (Blum and Rivest, 1989). Despite the worst-case pessimism, local search algorithms such as gradient descent are very successful in practice. With some additional assumptions, many works tried to design algorithms that provably learn a neural network (Goel et al., 2016; Sedghi and Anandkumar, 2014; Janzamin et al., 2015). However these algorithms are not gradient-based and do not provide insight on why local search algorithm works well.

Focusing on gradient-based algorithms, a line of research (Tian, 2017; Brutzkus and Globerson, 2017; Zhong et al., 2017a;b; Li and Yuan, 2017; Du et al., 2017b;c) analyzed the behavior of (stochastic) gradient descent with a structural assumption on the input distribution. The major drawback of these papers is that they all focus on the regression setting with least-squares loss and further assume the model is realizable meaning the label is the output of a neural network plus a zero mean noise, which is unrealistic. In the case of more than one hidden unit, the papers of (Li and Yuan, 2017; Zhong et al., 2017b) further require a stringent initialization condition to recover the true parameters.

Finding the optimal weights of a neural network is nonconvex problem. Recently, researchers found that if the objective functions satisfy the following two key properties: (1) all local minima are global and (2) all saddle points and local maxima are strict, then first order method like gradient descent (Ge et al., 2015; Jin et al., 2017; Levy, 2016; Du et al., 2017a; Lee et al., 2016) can find a global minimum.

This motivates the research of studying the landscape of neural networks (Kawaguchi, 2016; Choromanska et al., 2015; Freeman and Bruna, 2016; Zhou and Feng, 2017; Nguyen and Hein, 2017a;b; Ge et al., 2017; Safran and Shamir, 2017; Soltanolkotabi et al., 2017; Poston et al., 1991; Haeffele and Vidal, 2015; Haeffele et al., 2014; Soudry and Hoffer, 2017) In particular, Haeffele and Vidal (2015); Poston et al. (1991); Nguyen and Hein (2017a;b)studied the effect of over-parameterization on training the neural networks. These results require a large amount of over-parameterization that the width of one of the hidden layers has to be greater than the number of training examples, which is unrealistic in commonly used neural networks. Re-

cently, Soltanolkotabi et al. (2017) showed for shallow neural networks, the number of hidden nodes is only required to be larger or equal to the input dimension for ℓ_2-loss. In comparison, our theorems work for general loss functions with regularization under the same assumption. Further we also propose a new form of over-parameterization, namely as long as $k \geq \sqrt{2n}$, the loss function also admits a benign landscape.

We now turn our attention to generalization ability of learned neural networks. It is well known that the classical learning theory cannot explain the generalization ability because VC-dimension of neural networks is large (Harvey et al., 2017; Zhang et al., 2016). A line of research tries to explain this phenomenon by studying the implicit regularization from stochastic gradient descent algorithm (Hardt et al., 2015; Pensia et al., 2018; Mou et al., 2017; Brutzkus et al., 2017; Li et al., 2017). However, the generalization bounds of these papers often depend on the number of epochs SGD runs, which is large in practice. Another direction is to study the generalization ability based on the norms of weight matrices in neural networks (Neyshabur et al., 2015; 2017a;b; Bartlett et al., 2017; Liang et al., 2017; Golowich et al., 2017; Dziugaite and Roy, 2017; Wu et al., 2017). Our theorem on generalization ability also uses this idea but is more specialized to the network architecture (1).

After the initial submission of this manuscript, we became aware of concurrent work of (Bhojanapalli et al., 2018), which also considered the smoothed analysis technique to solve semi-definite programs in penalty form. The mathematical techniques in our work and (Bhojanapalli et al., 2018) are similar, but the focus is on two distinct problems of solving semi-definite programs and quadratic activation neural networks.

2. Preliminaries

We use bold-faced letters for vectors and matrices. For a vector $\mathbf{v}$, we use $\|\mathbf{v}\|_2$ to denote the Euclidean norm. For a matrix $\mathbf{M}$, we denote $\|\mathbf{M}\|_2$ the spectral norm and $\|\mathbf{M}\|_F$ the Frobenius norm. We let $\mathcal{N}(\mathbf{M})$ to denote the left null-space of $\mathbf{M}$, i.e.

$$\mathcal{N}(\mathbf{M}) = \left\{ \mathbf{v} : \mathbf{v}^\top \mathbf{M} = \mathbf{0} \right\}.$$

We use $\Sigma(M)$ to denote the set of matrices with Frobenius norm bounded by M and $\Sigma_1(1)$ to denote the set of rank-1 matrices with spectral norm bounded by 1. We also denote $\mathcal{S}^d$ the set of $d \times d$ symmetric positive semidefinite matrices.

In this paper, we characterize the landscape of over-parameterized neural networks. More specifically we study the properties of critical points of empirical loss. Here for a loss function $L(\mathbf{W})$, a critical point $\mathbf{W}^*$ satisfies $\nabla L(\mathbf{W}^*) = 0$. A critical point can be a local minimum or

a saddle point.[1] If $\mathbf{W}^*$ is a local minimum, then there is a neighborhood O around $\mathbf{W}^*$ such that $L(\mathbf{W}^*) \leq L(\mathbf{W})$ for all $\mathbf{W} \in O$. If $\mathbf{W}^*$ is a saddle point, then for all neighborhood O around $\mathbf{W}^*$, there is a $\mathbf{W} \in O$ such that $L(\mathbf{W}) < L(\mathbf{W}^*)$.

Ideally, we would like a loss function that satisfies the following two geometric properties.

Property 2.1 (All local minima are global). *If $\mathbf{W}^*$ is a local minimum of $L(\cdot)$ it is also the global minimum, i.e., $\mathbf{W}^* \in \operatorname{argmin}_{\mathbf{W}} L(\mathbf{W})$.*

Property 2.2 (All saddles are strict). *At a saddle point $\mathbf{W}_s$, there is a direction $\mathbf{U} \in \mathbb{R}^{k \times d}$ such that*

$$vect(\mathbf{U})^\top \nabla^2 L(\mathbf{W}_s) \, vect(\mathbf{U}) < 0.$$

If a loss function $L(\cdot)$ satisfies Property 2.1 and Property 2.2, recent algorithmic advances in non-convex optimization show randomly initialized gradient descent algorithm or perturbed gradient descent can find a global minimum (Lee et al., 2016; Ge et al., 2015; Jin et al., 2017; Du et al., 2017a).

Lastly, standard applications of Rademacher complexity theory will be used to derive generalization bounds.

Definition 2.1 (Definition of Rademacher Complexity). *Given a sample $S = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$, the empirical Rademacher complexity of a function class $\mathcal{F}$ is defined as*

$$R_S(\mathcal{F}) = \frac{1}{n} \mathbb{E}_\sigma \left[\sup_{f \in \mathcal{F}} \sum_{i=1}^n \sigma_i f(\mathbf{x}_i) \right]$$

where $\sigma = (\sigma_1, \ldots, \sigma_m)$ are independent random variables drawn from the Rademacher distribution, i.e., $\mathbb{P}(\sigma_i = 1) = \mathbb{P}(\sigma_i = -1) = 1/2$ for $i = 1, \ldots, n$.

3. Overparametrization Helps Optimization

In this section we present our main results on explaining why over-parametrization helps local search algorithms find a global optimal solution. We consider two kinds of over-parameterization, $k \geq d$ and $\frac{k(k+1)}{2} > n$. We begin with the simpler case when $k \geq d$.

Theorem 3.1. *Assume we have an arbitrary data set of input/label pairs $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$ for $i = 1, \ldots, n$ and a convex C^2 loss $\ell(\hat{y}, y)$. Consider a neural network of the form $f(\mathbf{x}, \mathbf{W}) = \sum_{j=1}^k \sigma(\mathbf{w}_j^\top \mathbf{x})$ with $\sigma(z) = z^2$ and $\mathbf{W} \in \mathbb{R}^{k \times d}$ denoting the weights connecting input to hidden layers. Suppose $k \geq d$. Then, the training loss as a*

function of weight $\mathbf{W}$ of the hidden layers

$$L(\mathbf{W}) = \frac{1}{2n} \ell(f(\mathbf{x}_i, \mathbf{W}), y_i) + \frac{\lambda}{2} \|\mathbf{W}\|_F^2$$

obeys Property 2.1 and Property 2.2.

The above result states that given an *arbitrary* data set, the optimization landscape has benign properties that facilitate finding globally optimal neural networks. In particular, by setting the last layer to be the average pooling layer, all local minima are global minima and all saddles have a direction of negative curvature. This in turn implies that gradient descent on the first layer weights, when initialized at random, converges to a global optimum. These desired properties hold as long as the hidden layer is wide ($k \geq d$).

An interesting and perhaps surprising aspect of Theorem 3.1 is its generality. It applies to arbitrary data set of any size with any convex differentiable loss function.

Now we consider the second case when $\frac{k(k+1)}{2} > n$. As mentioned earlier, in practice this is often a milder requirement than $k \geq d$, and one of the main novelties of this paper.

Theorem 3.2. *Assume we have an arbitrary data set of input/label pairs $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$ for $i = 1, \ldots, n$, and a convex C^2 loss $\ell(\hat{y}, y)$. Consider a neural network of the form $f(\mathbf{x}) = \sum_{j=1}^k \sigma(\mathbf{w}_j^\top \mathbf{x})$ with $\sigma(z) = z^2$ and $\mathbf{W} \in \mathbb{R}^{k \times d}$ denoting the weights connecting input to hidden layers. Suppose $\frac{k(k+1)}{2} > n, k < d$ and $\mathbf{C}$ is a random positive semidefinite matrix with $\|\mathbf{C}\|_F \leq \delta$ whose distribution is absolutely continuous with respect to Lebesgue measure. Then, the training loss as a function of weight $\mathbf{W}$ of the hidden layers*

$$L(\mathbf{W}) = \frac{1}{2n} \ell(f(\mathbf{x}_i), y_i) + \frac{\lambda}{2} \|\mathbf{W}\|_F^2 + \langle \mathbf{C}, \mathbf{W}^\top \mathbf{W} \rangle \tag{4}$$

obeys Property 2.1 and Property 2.2. Further, any global optimal solution $\widehat{\mathbf{W}}$ of Problem (4) satisfies

$$\frac{1}{2n} \sum_{i=1}^n \ell\left(f(\mathbf{x}_i, \widehat{\mathbf{W}}), y_i\right) + \frac{\lambda}{2} \left\|\widehat{\mathbf{W}}\right\|_F^2$$
$$\leq \frac{1}{2n} \sum_{i=1}^n \ell(f(\mathbf{x}_i, \mathbf{W}^*), y_i) + \frac{\lambda}{2} \|\mathbf{W}^*\|_2^F + \delta \|\mathbf{W}^*\|_F^2$$

where

$$\mathbf{W}^* \in \operatorname{argmin}_{\mathbf{W}} \sum_{i=1}^n \frac{1}{2n} \ell(f(\mathbf{x}_i, \mathbf{W}), y_i) + \frac{\lambda}{2} \|\mathbf{W}^*\|_F^2.$$

Similar to Theorem 3.1, Theorem 3.2 states that if $\frac{k(k+1)}{2} > n$, then for an arbitrary data set, the *perturbed* objective

function (3) has the desired properties that enable local search heuristics to find globally optimal solution for a general class of loss functions. Further, we can choose this perturbation to be arbitrarily small so the minimum of (3) is close to (2).

The proof of theorem is inspired by a line of literature started by Pataki (1998; 2000); Burer and Monteiro (2003); Boumal et al. (2016). In summary, Boumal et al. (2016) showed that for "almost all" semidefinite programs, every local minima of the rank r non-convex formulation of an SDP is a global minimum of the original SDP. However, this theorem applies with the important caveat of only applying to semidefinite programs that do not fall into a measure zero set. Our primary contribution is to develop a procedure that exploits this by a) constructing a perturbed objective to avoid the measure zero set, b) proving that the perturbed objective has Property 2.1 and 2.2, and c) showing the optimal value of the perturbed objective is close to the original objective. Further note that the analysis of (Boumal et al., 2016) does not apply since our loss functions, such as the logistic loss, are not semi-definite representable. We refer readers to Section 5.2 for more technical insights.

4. Weight-decay Helps Generalization

In this section we switch our focus to the generalization ability of the learned neural network. Since we use weight-decay or equivalently ℓ_2 regularization in (2), the Frobenius norm of learned weight $\mathbf{W}$ is bounded. Therefore, in this section we focus weight matrix in bounded Frobenius norm space, i.e., $\|\mathbf{W}\|_F \in \Sigma(M)$.

To derive the generalization bound, we first recall the classical generalization bound based on Rademacher complexity bound (c.f. Theorem 2 of (Koltchinskii and Panchenko, 2002)).

Theorem 4.1. *Assume each data point is sampled i.i.d from some distribution $\mathcal{P}$, i.e.,*

$$(\mathbf{x}_i, y_i) \sim P \text{ for } i = 1, \ldots, n.$$

We denote $S = \{\mathbf{x}_i, y_i\}_{i=1}^n$ and $L_{tr}(\mathbf{W}) = \frac{1}{n} \sum_{i=1}^n \ell(f(\mathbf{x}_i, \mathbf{W}), y_i)$ and $L_{te}(\mathbf{W}) = \mathbb{E}_{(\mathbf{x},y) \sim \mathcal{P}}[\ell(f(\mathbf{x}_i, \mathbf{W}), y_i)]$. Suppose loss function $\ell(\cdot, \cdot)$ is L-Lipschitz in the first argument, then for all $\mathbf{W} \in \Sigma(M)$, we have with high probability

$$L_{te}(\mathbf{W}) - L_{tr}(\mathbf{W}) \leq CL \cdot R_S(\Sigma(M))$$

where $C > 0$ is an absolute constant and $R_S(\Sigma(M))$ is the Rademacher complexity of $\Sigma(M)$.

With Theorem 4.1 at hand, we only need to bound the Rademacher complexity of $\Sigma(M)$. Note that Rademacher complexity is a distribution dependent quantity. If the data

is arbitrary, we cannot have any guarantee. We begin with a theorem for bounded input domain.

Theorem 4.2. *Suppose input is sampled from a bounded ball in $\mathcal{R}^d$, i.e., $\|\mathbf{x}\|_2 \leq b$ for some $b > 0$ and $\mathbb{E}\left[\|\mathbf{x}\mathbf{x}^\top\|_2^2\right] \leq B$, then the Rademacher complexity satisfies*

$$R_S(\Sigma(M)) \leq \sqrt{\frac{2b^4 M^4 \log d}{n}}.$$

Combining Theorem 4.1 and Theorem 4.2 we can obtain a generalization bound.

Theorem 4.3. *Under the same assumptions of Theorem 4.1 and Theorem 4.2, we have*

$$L_{te}(\mathbf{W}) - L_{tr}(\mathbf{W}) \leq CLM^2 b^2 \sqrt{\frac{\log d}{n}}.$$

for some absolute constant $C > 0$.

While Theorem 4.2 is a valid bound, it is rather pessimistic because we only assume $\mathbf{x}$ is bounded. Consider the following scenario in which each input is sampled from a standard Gaussian distribution $\mathbf{x}_i \sim N(0, \mathbf{I})$. Then ignoring the logarithmic factors, using standard Gaussian concentration bound we can show with high probability $\|\mathbf{x}_i\|_2 = \widetilde{O}\left(\sqrt{d}\right)$. [2] Plugging in this bound we have

$$L_{te}(\mathbf{W}) - L_{tr}(\mathbf{W}) \leq CLM^2 \sqrt{\frac{d^2 \log d}{n}} \tag{5}$$

Note in this bound, it has a quadratic dependency on the dimension, so we need to have $n = \Omega(d^2)$ to have a meaningful bound.

In fact, for specific distributions like Gaussian using Theorem 5.2, we can often derive a stronger generalization bound.

Corollary 4.1. *Suppose $\mathbf{x}_i \sim N(0, \mathbf{I})$ for $i = 1, \ldots, n$. If the number of samples satisfies $n \geq d \log d$, we have with high probability that Rademacher complexity satisfies*

$$R_S(\Sigma(M)) \leq C\sqrt{\frac{M^4 d}{n}}.$$

for some absolute constant $C > 0$.

Again, combining Theorem 4.1 and Corollary 4.1 we obtain the following generalization bound for Gaussian input distribution

Theorem 4.4. *Under the same assumptions of Theorem 4.1 and Corollary 4.1, we have*

$$L_{te}(\mathbf{W}) - L_{tr}(\mathbf{W}) \leq CLM^2 \sqrt{\frac{d}{n}}.$$

for some absolute constant $C > 0$.

---

[2] $\widetilde{O}(\cdot)$ hides logarithmic factors.

Comparing Theorem 4.4 with generalization bound (5), Theorem 4.4 has an $O(\sqrt{d})$ advantage. Theorem 4.4 has the $\sqrt{d/n}$ dependency, which is the usual parametric rate. Further in practice, number of training samples and input dimension are often of the same order for common datasets and architectures (Zhang et al., 2016).

Corollary 4.1 is a special case of the more general Theorem 5.2 which only requires a bound on the fourth moment, $\left\|\sum_{i=1}^{n}(x_i x_i^\top)^2\right\|_2 \leq s$. In general, our theorems suggest if the Frobenius norm of weight matrix $\mathbf{W}$ is small and the input $\mathbf{x}$ is sampled from a benign distribution with controlled 4th moments, then we have good generalization.

As a concrete scenario, consider a favorable setting where the true data can be correctly classified by a small network using only $k_0 \ll k$ hidden units. The weights W^* are non-zero only in the first k_0 rows and $\max_{j\in[k_0]} \left\|e_j^\top W^*\right\|_2 \leq R$. From Theorem 4.4 to reach generalization gap of ϵ, we have sample complexity of $n \geq \frac{1}{\epsilon^2} C^2 L^2 R^4 d k_0^2$, which only depends on the effective number of hidden units $k_0 \ll k$. The same result can be reached for more general input distributions by using Theorem 5.2 in place of Theorem 4.4.

5. Proofs

5.1. Proof of Theorem 3.1 and Theorem 3.2

Our proofs of over-parametrization helps optimization build upon existing geometric characterization on matrix factorization. We first cite a useful Theorem by Haeffele et al. (2014).[3]

Theorem 5.1 (Theorem 2 of (Haeffele et al., 2014) adapted to our setting)**.** *Let $\ell(\cdot, \cdot)$ be a twice differentiable convex function in the first argument. If the function $L(\mathbf{W})$ defined in (2) at a rank-deficient matrix $\mathbf{W}$ satisfies*

$$\nabla L(\mathbf{W}) = \mathbf{0} \text{ and } \nabla^2 L(\mathbf{W}) \succcurlyeq \mathbf{0},$$

then $\mathbf{W}$ is a global minimum.

Proof of Theorem 3.1. We prove Property 2.1 and Property 2.2 simultaneously by showing if a $\mathbf{W}$ satisfy

$$\nabla L(\mathbf{W}) = \mathbf{0} \text{ and } \nabla^2 L(\mathbf{W}) \succcurlyeq 0$$

then it is a global minimum.

If $\operatorname{rank}(\mathbf{W}) < d$, we can directly apply Theorem 5.1. Thus it remains to consider the case $\operatorname{rank}(\mathbf{W}) = d$. We first

---

[3] Theorem 2 of (Haeffele et al., 2014) assumes $\mathbf{W}$ is a local minimum, but scrutinizing its proof, we can see that the assumption can be relaxed to $\nabla L(\mathbf{W}) = \mathbf{0}$ and $\nabla^2 L(\mathbf{W}) \succcurlyeq \mathbf{0}$.

notice that $\nabla L(\mathbf{W}) = 0$ is equivalent to

$$\mathbf{W}\left(\frac{1}{n}\sum_{i=1}^{n}\frac{\partial\ell(\hat{y}_i, y_i)}{\partial\hat{y}_i}\mathbf{x}_i\mathbf{x}_i^\top + \lambda\mathbf{I}\right) = \mathbf{0} \qquad (6)$$

where $\hat{y}_i = f(\mathbf{x}_i, \mathbf{W})$. Since $\operatorname{rank}(\mathbf{W}) = d$ and $k \geq d$, we know $\mathbf{W}$ has a left pseudo-inverse, i.e., there exists $\mathbf{W}^\dagger$ such that $\mathbf{W}^\dagger\mathbf{W} = \mathbf{I}$. Multiplying $\mathbf{W}^\dagger$ on the left in Equation (6), we have

$$\frac{1}{n}\sum_{i=1}^{n}\frac{\partial\ell(\hat{y}_i, y_i)}{\partial\hat{y}_i}\mathbf{x}_i\mathbf{x}_i^\top + \lambda\mathbf{I} = \mathbf{0}. \qquad (7)$$

To prove Theorem 3.1, the key idea is to consider the follow reference optimization problem.

$$\min_{\mathbf{M}:\mathbf{M}\in\mathcal{S}^d} L(\mathbf{M}) = \frac{1}{n}\sum_{i=1}^{n}\ell\left(\mathbf{x}_i^\top\mathbf{M}\mathbf{x}_i, y_i\right) + \lambda\left\|\mathbf{M}\right\|_*. \quad (8)$$

Problem (8) is a convex optimization problem in $\mathbf{M}$ and has the same global minimum as the original problem. Now we denote $\tilde{y}_i = \mathbf{x}_i^\top\mathbf{M}\mathbf{x}_i$. Since this is a convex function, the first-order optimality condition for global optimality is

$$\mathbf{0} \in \frac{1}{n}\sum_{i=1}^{n}\frac{\partial\ell(\tilde{y}_i, y_i)}{\partial\tilde{y}_i}\mathbf{x}_i\mathbf{x}_i^\top + \lambda\partial\left\|\mathbf{M}\right\|_*,$$

$\mathbf{M}$ is a global minimum.

Using Equation (7), we know $\mathbf{M} = \mathbf{W}^\top\mathbf{W}$ achieves the global minimum in Problem (8). The proof is thus complete. $\square$

5.2. Proof of Theorem 3.2

Proof. We first prove $L_{\mathbf{C}}(\mathbf{W})$ satisfies Property 2.1 and Property 2.2. Similar to the proof of Theorem 3.1, we prove these two properties simultaneously by showing if a $\mathbf{W}$ satisfy

$$\nabla L_{\mathbf{C}}(\mathbf{W}) = \mathbf{0} \text{ and } \nabla^2 L_{\mathbf{C}}(\mathbf{W}) \succcurlyeq 0 \qquad (9)$$

then it is a global minimum. Because of Theorem 5.1, we only need to show that if $\mathbf{W}$ satisfy condition (9), it is rank deficient, i.e. $\operatorname{rank}(\mathbf{W}) < k$.

For the gradient condition, we have

$$\mathbf{W}\left(\frac{1}{n}\sum_{i=1}^{n}\frac{\partial\ell(\hat{y}_i, y_i)}{\partial\hat{y}_i}\mathbf{x}_i\mathbf{x}_i^\top + \lambda\mathbf{I}\right) + \mathbf{W}\mathbf{C} = 0.$$

For simplicity we denote $v_i = \frac{\partial\ell(\hat{y}_i, y_i)}{\partial\hat{y}_i}$ where $\hat{y}_i = \mathbf{x}_i^\top\mathbf{W}^\top\mathbf{W}\mathbf{x}_i$ and $\mathbf{S}(\mathbf{v}) = \sum_{i=1}^{n}v_i\mathbf{x}_i\mathbf{x}_i^\top$. Using the first order condition we know $\mathbf{W}$ is in the null space of $(\mathbf{S}(\mathbf{v}) + \lambda\mathbf{I} + \mathbf{C})$. Thus, we can bound the rank of $\mathbf{W}$ by

$$\operatorname{rank}(\mathbf{W}) \leq \dim\mathcal{N}(\mathbf{S}(\mathbf{v}) + \lambda\mathbf{I} + \mathbf{C})$$
$$\leq \max_{\mathbf{v}}\dim\mathcal{N}(\mathbf{S}(\mathbf{v}) + \lambda\mathbf{I} + \mathbf{C}).$$

We prove by contradiction. Assume $\text{rank}(\mathbf{W}) \geq k$, we must have

$$k \leq \max_{\mathbf{v}} \mathcal{N}\left(\mathbf{S}(\mathbf{v}) + \lambda \mathbf{I} + \mathbf{C}\right).$$

Now define $\mathbf{M} = \mathbf{S}(\mathbf{v}^*) + \lambda \mathbf{I} + \mathbf{C}$ with

$$\mathbf{v}^* = \text{argmax} \dim \mathcal{N}\left(\mathbf{S}(\mathbf{v}) + \lambda \mathbf{I} + \mathbf{C}\right).$$

Thus we have following conditions

$$\mathbf{C} = \mathbf{M} - \mathbf{S}(\mathbf{v}^*) - \lambda \mathbf{I},$$
$$\dim \mathcal{N}(\mathbf{M}) \geq k.$$

The key idea is to use these two conditions to upper bound the dimension of $\mathbf{C}$. To this end, we first define the set

$$\mathcal{B}_\ell = \left\{\mathbf{A} : \mathbf{A} = \mathbf{M} - \mathbf{S}(\mathbf{v}) - \lambda \mathbf{I}, \mathbf{M} \in \mathcal{S}^d, \mathbf{v} \in \mathbb{R}^n, \right.$$
$$\left. \dim \mathcal{N}(\mathbf{M} - \lambda \mathbf{I}) = \ell \right\}.$$

Note that the dimension of the manifold $\mathcal{B}_\ell$ is

$$\left(\frac{d(d+1)}{2} - \frac{\ell(\ell+1)}{2}\right) + n$$

where the first term is the dimension of $d \times d$ matrices, the second term is the dimension of the null space and the last term is dimension of $\text{Range}(\mathbf{S}(\mathbf{v}))$ for $\mathbf{v} \in \mathbb{R}^n$, which is upper bounded by n.

Next note that $\mathcal{B}_{\ell_1} \subset \mathcal{B}_{\ell_2}$ for $\ell_1 \geq \ell_2$. Therefore, we can compute the dimension of the union

$$\dim\left(\cup_{\ell=k}^d \mathcal{B}_\ell\right) = \left(\frac{d(d+1)}{2} - \frac{k(k+1)}{2}\right) + n.$$

Note because we assume $\frac{k(k+1)}{2} > n$, we have $\dim\left(\cup_{\ell=k}^d \mathcal{B}_\ell\right) < \frac{d(d+1)}{2}$. However, recall $\mathbf{C} \in \cup_{\ell=k}^d \mathcal{B}_\ell$ by definition, so we have $\mathbf{C}$ is sampled from a low-dimensional manifold which has Lebesuge measure 0. Since we sample $\mathbf{C}$ from a distribution that is absolute continuous with respect to the Lebesgue measure, the event $\left\{\mathbf{C} \in \cup_{\ell=k}^d \mathcal{B}_\ell\right\}$ happens with probability 0. Therefore, with probability 1, $\text{rank}(\mathbf{W}) < k$. The proof of the first part of Theorem 3.2 is complete.

For the second part. Let $\widehat{\mathbf{W}} = \text{argmin}_{\mathbf{W}} L_{\mathbf{C}}(\mathbf{W})$ and $\mathbf{W}^* = \text{argmin}_{\mathbf{W}}(\mathbf{W})$. Therefore we have

$$L\left(\widehat{\mathbf{W}}\right) + \langle \mathbf{C}, \widehat{\mathbf{W}}^\top \widehat{\mathbf{W}} \rangle$$
$$\leq L(\mathbf{W}^*) + \langle \mathbf{C}, (\mathbf{W}^*)^\top \mathbf{W}^* \rangle$$
$$\leq L(\mathbf{W}^*) + \delta \|\mathbf{W}^*\|_F^2.$$

Note because $\mathbf{C}$ and $\widehat{\mathbf{W}}^\top \widehat{\mathbf{W}}$ are both positive semidefinite, we have $\langle \mathbf{C}, \widehat{\mathbf{W}}^\top \widehat{\mathbf{W}} \rangle \geq 0$. Thus $L\left(\widehat{\mathbf{W}}\right) \leq L(\mathbf{W}^*) + \delta \|\mathbf{W}^*\|_F^2$. □

5.3. Proof of Theorem 4.2 and Theorem 4.4

Our proof is inspired by (Srebro and Shraibman, 2005) which exploits the structure of nuclear norm bounded space. We first prove a general Theorem that only depends on the property of the fourth-moment of input random variables.

Theorem 5.2. *Suppose the input random variable satisfies* $\left\|\sum_{i=1}^n \left(\mathbf{x}_i \mathbf{x}_i^\top\right)^2\right\|_2 \leq s$. *Then the Rademacher complexity of* $\Sigma(M)$ *is bounded by*

$$R_S(\Sigma(M)) \leq \frac{\sqrt{2M^4 s \log d}}{n}.$$

Proof. For a given set of inputs $S = \{\mathbf{x}_i\}_{i=1}^n$ in our context, we can write Rademacher complexity as

$$R_S(\Sigma(M)) = \frac{1}{n} \mathbb{E}_\sigma \left[\sup_{\mathbf{W} \in \Sigma(M)} \sum_{i=1}^n \sigma_i \mathbf{x}_i^\top \mathbf{W}^\top \mathbf{W} \mathbf{x}_i\right]$$

Since Rademacher complexity does not change when taking convex combinations, we can first bound Rademacher complexity of the class of rank-1 matrices with spectral norm bounded by 1 and then take convex hull and scale by M. Note for $\mathbf{W} \in \Sigma_1(1)$, we can write $\mathbf{W} = \mathbf{v}\mathbf{w}^\top$ with $\|\mathbf{w}\|_2 \leq 1$ and $\|\mathbf{v}\|_2 = 1$. Using this expression, we can obtain an explicit formula of Rademacher complexity.

$$R_S(\Sigma_1(1))$$
$$= \frac{1}{n} \mathbb{E}_\sigma \left[\sup_{\mathbf{W} \in \Sigma_1(1)} \sum_{i=1}^n \sigma_i \mathbf{x}_i^\top \mathbf{W}^\top \mathbf{W} \mathbf{x}_i\right]$$
$$= \frac{1}{n} \mathbb{E}_\sigma \left[\sup_{\mathbf{W} \in \Sigma_1(1)} \sum_{i=1}^n \sigma_i \mathbf{x}_i^\top \left(\mathbf{w}^\top \mathbf{v}\right)^\top \left(\mathbf{v}\mathbf{w}^\top\right) \mathbf{x}_i\right]$$
$$= \frac{1}{n} \mathbb{E}_\sigma \left[\sup_{\mathbf{w}: \|\mathbf{w}\|_2 \leq 1} \sum_{i=1}^n \sigma_i \mathbf{x}_i^\top \mathbf{w}\mathbf{w}^\top \mathbf{x}_i\right]$$
$$= \frac{1}{n} \mathbb{E}_\sigma \left[\sup_{\mathbf{w}: \|\mathbf{w}\|_2 \leq 1} \sum_{i=1}^n \sigma_i \mathbf{w}^\top \mathbf{x}_i^\top \mathbf{x}_i \mathbf{w}\right]$$
$$= \frac{1}{n} \mathbb{E}_\sigma \left[\left\|\sum_{i=1}^n \sigma_i \mathbf{x}_i \mathbf{x}_i^\top\right\|_2\right].$$

Now, to bound $\mathbb{E}_\sigma \left[\left\|\sum_{i=1}^n \sigma_i \mathbf{x}_i \mathbf{x}_i^\top\right\|_2\right]$, we can use the results from random matrix theory on Rademacher series. Recall that we assume

$$\left\|\sum_{i=1}^n \left(\mathbf{x}_i \mathbf{x}_i^\top\right)^2\right\|_2 \leq s$$

and notice that

$$\mathbb{E}_\sigma \left[\sum_{i=1}^n \sigma_i \mathbf{x}_i \mathbf{x}_i^\top\right] = \mathbf{0}.$$

Applying Rademacher matrix series expectation bound (Theorem 4.6.1 of (Tropp et al., 2015)), we have

$$\mathbb{E}_\sigma\left[\left\|\sum_{i=1}^n \sigma_i \mathbf{x}_i \mathbf{x}_i^\top\right\|_2\right] \leq \sqrt{2\left\|\sum_{i=1}^n \left(\mathbf{x}_i \mathbf{x}_i^\top\right)^2\right\|_2 \log d}$$
$$\leq \sqrt{2s \log d}.$$

Now taking the convex hull and, scaling by M we obtain the desired result. $\square$

With Theorem 5.2 at hand, for different distributions, we only need to bound $\left\|\sum_{i=1}^n \left(\mathbf{x}_i \mathbf{x}_i^\top\right)^2\right\|_2$.

Proof of Theorem 4.3. Since we assume $\|\mathbf{x}_i\|_2 \leq b$, we directly have

$$\left\|\sum_{i=1}^n \left(\mathbf{x}_i \mathbf{x}_i^\top\right)^2\right\|_2 \leq \sum_{i=1}^n \left\|\left(\mathbf{x}_i \mathbf{x}_i^\top\right)^2\right\|_2$$
$$= \sum_{i=1}^n \|\mathbf{x_i}\|_2^2 \left\|\mathbf{x}_i \mathbf{x}_i^\top\right\|_2$$
$$\leq nb^4.$$

Plugging this bound in Theorem 5.2 we obtain the desired inequality. $\square$

Proof of Corollary 4.1 and Theorem 4.4. To prove Corollary 4.1, we use Theorem 5.2 and Lemma 4.7 in (Soltanolkotabi et al., 2017) to upper bound $s = \left\|\sum_{i=1}^n \|x_i\|^2 x_i x_i^\top\right\|$. By letting $A = I$ in Lemma 4.7, we find that

$$s \leq Cnd,$$

with probability at least $1 - \frac{C}{d}$. This completes the proof of Corollary 4.1.

Using this bound in Theorem 5.2 comletes the proof of Theorem 4.4. $\square$

6. Conclusion and Future Works

In this paper we provided new theoretical results on over-parameterized neural networks. Using smoothed analysis, we showed as long as the number of hidden nodes is bigger than the input dimension *or* square root of the number of training data, the loss surface has benign properties that enable local search algorithms to find global minima. We further use the theory of Rademacher complexity to show the learned neural can generalize well.

Our next step is consider neural networks with other activation functions and how over-parametrization allows for efficient local-search algorithms to find near global minimzers. Another interesting direction to extend our results to deeper model.

References

Andrew R Barron. Approximation and estimation bounds for artificial neural networks. *Machine learning*, 14(1): 115–133, 1994.

Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, pages 6241–6250, 2017.

Srinadh Bhojanapalli, Nicolas Boumal, Prateek Jain, and Praneeth Netrapalli. Smoothed analysis for low-rank solutions to semidefinite programs in quadratic penalty form. *arXiv preprint arXiv:1803.00186*, 2018.

Avrim Blum and Ronald L Rivest. Training a 3-node neural network is NP-complete. In *Advances in neural information processing systems*, pages 494–501, 1989.

Helmut Bölcskei, Philipp Grohs, Gitta Kutyniok, and Philipp Petersen. Optimal approximation with sparsely connected deep neural networks. *arXiv preprint arXiv:1705.01714*, 2017.

Nicolas Boumal, Vlad Voroninski, and Afonso Bandeira. The non-convex burer-monteiro approach works on smooth semidefinite programs. In *Advances in Neural Information Processing Systems*, pages 2757–2765, 2016.

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

Alon Brutzkus, Amir Globerson, Eran Malach, and Shai Shalev-Shwartz. Sgd learns over-parameterized networks that provably generalize on linearly separable data. *arXiv preprint arXiv:1710.10174*, 2017.

Samuel Burer and Renato DC Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 95(2):329–357, 2003.

Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In *Artificial Intelligence and Statistics*, pages 192–204, 2015.

Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. *arXiv preprint arXiv:1612.08083*, 2016.

Simon S Du, Chi Jin, Jason D Lee, Michael I Jordan, Barnabas Poczos, and Aarti Singh. Gradient descent can take exponential time to escape saddle points. *arXiv preprint arXiv:1705.10412*, 2017a.

Simon S Du, Jason D Lee, and Yuandong Tian. When is a convolutional filter easy to learn? *arXiv preprint arXiv:1709.06129*, 2017b.

Simon S Du, Jason D Lee, Yuandong Tian, Barnabas Poczos, and Aarti Singh. Gradient descent learns one-hidden-layer cnn: Don't be afraid of spurious local minima. *arXiv preprint arXiv:1712.00779*, 2017c.

Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 2017.

C Daniel Freeman and Joan Bruna. Topology and geometry of half-rectified network optimization. *arXiv preprint arXiv:1611.01540*, 2016.

Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points − online stochastic gradient for tensor decomposition. In *Proceedings of The 28th Conference on Learning Theory*, pages 797–842, 2015.

Rong Ge, Jason D Lee, and Tengyu Ma. Learning one-hidden-layer neural networks with landscape design. *arXiv preprint arXiv:1711.00501*, 2017.

Surbhi Goel, Varun Kanade, Adam Klivans, and Justin Thaler. Reliably learning the ReLU in polynomial time. *arXiv preprint arXiv:1611.10258*, 2016.

Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. *arXiv preprint arXiv:1712.06541*, 2017.

Benjamin Haeffele, Eric Young, and Rene Vidal. Structured low-rank matrix factorization: Optimality, algorithm, and applications to image processing. In *International Conference on Machine Learning*, pages 2007–2015, 2014.

Benjamin D Haeffele and René Vidal. Global optimality in tensor factorization, deep learning, and beyond. *arXiv preprint arXiv:1506.07540*, 2015.

Moritz Hardt, Benjamin Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. *arXiv preprint arXiv:1509.01240*, 2015.

Nick Harvey, Chris Liaw, and Abbas Mehrabian. Nearly-tight vc-dimension bounds for piecewise linear neural networks. *arXiv preprint arXiv:1703.02930*, 2017.

Majid Janzamin, Hanie Sedghi, and Anima Anandkumar. Beating the perils of non-convexity: Guaranteed training of neural networks using tensor methods. *arXiv preprint arXiv:1506.08473*, 2015.

Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M. Kakade, and Michael I. Jordan. How to escape saddle points efficiently. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1724–1732, 2017.

Kenji Kawaguchi. Deep learning without poor local minima. In *Advances In Neural Information Processing Systems*, pages 586–594, 2016.

Vladimir Koltchinskii and Dmitry Panchenko. Empirical margin distributions and bounding the generalization error of combined classifiers. *Annals of Statistics*, pages 1–50, 2002.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

Jason D Lee, Max Simchowitz, Michael I Jordan, and Benjamin Recht. Gradient descent only converges to minimizers. In *Conference on Learning Theory*, pages 1246–1257, 2016.

Kfir Y Levy. The power of normalization: Faster evasion of saddle points. *arXiv preprint arXiv:1611.04831*, 2016.

Yuanzhi Li and Yang Yuan. Convergence analysis of two-layer neural networks with ReLU activation. *arXiv preprint arXiv:1705.09886*, 2017.

Yuanzhi Li, Tengyu Ma, and Hongyang Zhang. Algorithmic regularization in over-parameterized matrix recovery. *arXiv preprint arXiv:1712.09203*, 2017.

Tengyuan Liang, Tomaso Poggio, Alexander Rakhlin, and James Stokes. Fisher-rao metric, geometry, and complexity of neural networks. *arXiv preprint arXiv:1711.01530*, 2017.

Roi Livni, Shai Shalev-Shwartz, and Ohad Shamir. On the computational efficiency of training neural networks. In *Advances in Neural Information Processing Systems*, pages 855–863, 2014.

Wenlong Mou, Liwei Wang, Xiyu Zhai, and Kai Zheng. Generalization bounds of sgld for non-convex learning: Two theoretical viewpoints. *arXiv preprint arXiv:1707.05947*, 2017.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *Conference on Learning Theory*, pages 1376–1401, 2015.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1707.09564*, 2017a.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems*, pages 5949–5958, 2017b.

Quynh Nguyen and Matthias Hein. The loss surface of deep and wide neural networks. *arXiv preprint arXiv:1704.08045*, 2017a.

Quynh Nguyen and Matthias Hein. The loss surface and expressivity of deep convolutional neural networks. *arXiv preprint arXiv:1710.10928*, 2017b.

Gábor Pataki. On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues. *Mathematics of operations research*, 23(2):339–358, 1998.

Gábor Pataki. The geometry of semidefinite programming. In *Handbook of semidefinite programming*, pages 29–65. Springer, 2000.

Ankit Pensia, Varun Jog, and Po-Ling Loh. Generalization error bounds for noisy, iterative algorithms. *arXiv preprint arXiv:1801.04295*, 2018.

Timothy Poston, C-N Lee, Y Choie, and Yonghoon Kwon. Local minima and back propagation. In *Neural Networks, 1991., IJCNN-91-Seattle International Joint Conference on*, volume 2, pages 173–176. IEEE, 1991.

Itay Safran and Ohad Shamir. Spurious local minima are common in two-layer relu neural networks. *arXiv preprint arXiv:1712.08968*, 2017.

Hanie Sedghi and Anima Anandkumar. Provable methods for training neural networks with sparse connectivity. *arXiv preprint arXiv:1412.2693*, 2014.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587): 484–489, 2016.

Mohammadreza Soltani and Chinmay Hegde. Towards provable learning of polynomial neural networks using low-rank matrix estimation. 2017.

Mahdi Soltanolkotabi, Adel Javanmard, and Jason D Lee. Theoretical insights into the optimization landscape of over-parameterized shallow neural networks. *arXiv preprint arXiv:1707.04926*, 2017.

Daniel Soudry and Elad Hoffer. Exponentially vanishing sub-optimal local minima in multilayer neural networks. *arXiv preprint arXiv:1702.05777*, 2017.

Nathan Srebro and Adi Shraibman. Rank, trace-norm and max-norm. In *International Conference on Computational Learning Theory*, pages 545–560. Springer, 2005.

Nathan Srebro, Jason Rennie, and Tommi S Jaakkola. Maximum-margin matrix factorization. In *Advances in neural information processing systems*, pages 1329–1336, 2005.

Matus Telgarsky. Benefits of depth in neural networks. *arXiv preprint arXiv:1602.04485*, 2016.

Yuandong Tian. An analytical formula of population gradient for two-layered ReLU network and its applications in convergence and critical point analysis. *arXiv preprint arXiv:1703.00560*, 2017.

Joel A Tropp et al. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015.

Thomas Wiatowski, Philipp Grohs, and Helmut Bölcskei. Energy propagation in deep convolutional neural networks. *IEEE Transactions on Information Theory*, 2017.

Lei Wu, Zhanxing Zhu, et al. Towards understanding generalization of deep learning: Perspective of loss landscapes. *arXiv preprint arXiv:1706.10239*, 2017.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

Kai Zhong, Zhao Song, and Inderjit S Dhillon. Learning non-overlapping convolutional neural networks with multiple kernels. *arXiv preprint arXiv:1711.03440*, 2017a.

Kai Zhong, Zhao Song, Prateek Jain, Peter L Bartlett, and Inderjit S Dhillon. Recovery guarantees for one-hidden-layer neural networks. *arXiv preprint arXiv:1706.03175*, 2017b.

Pan Zhou and Jiashi Feng. The landscape of deep learning algorithms. *arXiv preprint arXiv:1705.07038*, 2017.