Recovery Guarantees for One-hidden-layer Neural Networks*

Kai Zhong [1] **Zhao Song** [2] **Prateek Jain** [3] **Peter L. Bartlett** [4] **Inderjit S. Dhillon** [5]

Abstract

In this paper, we consider regression problems with one-hidden-layer neural networks (1NNs). We distill some properties of activation functions that lead to *local strong convexity* in the neighborhood of the ground-truth parameters for the 1NN squared-loss objective and most popular nonlinear activation functions satisfy the distilled properties, including rectified linear units (ReLUs), leaky ReLUs, squared ReLUs and sigmoids. For activation functions that are also smooth, we show *local linear convergence* guarantees of gradient descent under a resampling rule. For homogeneous activations, we show tensor methods are able to initialize the parameters to fall into the local strong convexity region. As a result, tensor initialization followed by gradient descent is guaranteed to recover the ground truth with sample complexity $d \cdot \log(1/\epsilon) \cdot \mathrm{poly}(k, \lambda)$ and computational complexity $n \cdot d \cdot \mathrm{poly}(k, \lambda)$ for smooth homogeneous activations with high probability, where d is the dimension of the input, k ($k \leq d$) is the number of hidden nodes, λ is a conditioning property of the ground-truth parameter matrix between the input layer and the hidden layer, ϵ is the targeted precision and n is the number of samples. To the best of our knowledge, this is the first work that provides recovery guarantees for 1NNs with both sample complexity and computational complexity *linear* in the input dimension and *logarithmic* in the precision.

[1]The University of Texas at Austin, zhongkai@ices.utexas.edu
[2]The University of Texas at Austin, zhaos@utexas.edu
[3]Microsoft Research, India, prajain@microsoft.com
[4]University of California, Berkeley, bartlett@cs.berkeley.edu
[5]The University of Texas at Austin, inderjit@cs.utexas.edu
*Full version is available at `https://arxiv.org/pdf/1706.03175`. Correspondence to: Kai Zhong <zhongkai@ices.utexas.edu>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017.

1. Introduction

Neural Networks (NNs) have achieved great practical success recently. Many theoretical contributions have been made very recently to understand the extraordinary performance of NNs. The remarkable results of NNs on complex tasks in computer vision and natural language processing inspired works on the expressive power of NNs (Cohen et al., 2016; Cohen & Shashua, 2016; Raghu et al., 2016; Daniely et al., 2016; Poole et al., 2016; Montufar et al., 2014; Telgarsky, 2016). Indeed, several works found NNs are very powerful and the deeper the more powerful. However, due to the high non-convexity of NNs, knowing the expressivity of NNs doesn't guarantee that the targeted functions will be learned. Therefore, several other works focused on the achievability of global optima. Many of them considered the over-parameterized setting, where the global optima or local minima close to the global optima will be achieved when the number of parameters is large enough, including (Freeman & Bruna, 2016; Haeffele & Vidal, 2015; Livni et al., 2014; Dauphin et al., 2014; Safran & Shamir, 2016; Hardt & Ma, 2017). This, however, leads to overfitting easily and can't provide any generalization guarantees, which are actually the essential goal in most tasks.

A few works have considered generalization performance. For example, (Xie et al., 2017) provide generalization bound under the Rademacher generalization analysis framework. Recently (Zhang et al., 2017a) describe some experiments showing that NNs are complex enough that they actually memorize the training data but still generalize well. As they claim, this cannot be explained by applying generalization analysis techniques, like VC dimension and Rademacher complexity, to classification loss (although it does not rule out a margins analysis—see, for example, (Bartlett, 1998); their experiments involve the unbounded cross-entropy loss).

In this paper, we don't develop a new generalization analysis. Instead we focus on parameter recovery setting, where we assume there are underlying ground-truth parameters and we provide recovery guarantees for the ground-truth parameters up to equivalent permutations. Since the parameters are exactly recovered, the generalization performance will also be guaranteed.

Several other techniques are also provided to recover the parameters or to guarantee generalization performance, such as tensor methods (Janzamin et al., 2015) and kernel methods (Arora et al., 2017). These methods require sample complexity $O(d^3)$ or computational complexity $\widetilde{O}(n^2)$, which can be intractable in practice.

Recently (Shamir, 2016) show that neither specific assumptions on the niceness of the input distribution or niceness of the target function alone is sufficient to guarantee learnability using gradient-based methods. In this paper, we assume data points are sampled from Gaussian distribution and the parameters of hidden neurons are linearly independent.

Our main contributions are as follows,

1. We distill some properties for activation functions, which are satisfied by a wide range of activations, including ReLU, squared ReLU, sigmoid and tanh. With these properties we show positive definiteness (PD) of the Hessian in the neighborhood of the ground-truth parameters given enough samples (Theorem 4.2). Further, for activations that are also smooth, we show local linear convergence is guaranteed using gradient descent.

2. We propose a tensor method to initialize the parameters such that the initialized parameters fall into the local positive definiteness area. Our contribution is that we reduce the sample/computational complexity from cubic dependency on dimension to linear dependency (Theorem 5.6).

3. Combining the above two results, we provide a globally converging algorithm (Algorithm 2) for smooth homogeneous activations satisfying the distilled properties. The whole procedure requires sample/computational complexity linear in dimension and logarithmic in precision (Theorem 6.1).

2. Related Work

The recent empirical success of NNs has boosted their theoretical analyses (Feng et al., 2016; Balduzzi, 2016; Balduzzi et al., 2016; Sagun et al., 2016; Andoni et al., 2014; Arora et al., 2017; Goel et al., 2017). In this paper, we classify them into three main directions.

2.1. Expressive Power

Expressive power is studied to understand the remarkable performance of neural networks on complex tasks. Although one-hidden-layer neural networks with sufficiently many hidden nodes can approximate any continuous function (Hornik, 1991), shallow networks can't achieve the same performance in practice as deep networks. Theoretically, several recent works show the depth of NNs plays an essential role in the expressive power of neural networks (Daniely et al., 2016). As shown in (Cohen et al., 2016; Cohen & Shashua, 2016; Telgarsky, 2016), functions that can be implemented by a deep network of polynomial size require exponential size in order to be implemented by a shallow network. (Raghu et al., 2016; Poole et al., 2016; Montufar et al., 2014; Arora et al., 2017) design some measures of expressivity that display an exponential dependence on the depth of the network. However, the increasing of the expressivity of NNs or its depth also increases the difficulty of the learning process to achieve a good enough model. In this paper, we focus on 1NNs and provide recovery guarantees using a finite number of samples.

2.2. Achievability of Global Optima

The global convergence is in general not guaranteed for NNs due to their non-convexity. It is widely believed that training deep models using gradient-based methods works so well because the error surface either has no local minima, or if they exist they need to be close in value to the global minima. (Swirszcz et al., 2016) present examples showing that for this to be true additional assumptions on the data, initialization schemes and/or the model classes have to be made. Indeed the achievability of global optima has been shown under many different types of assumptions.

In particular, (Choromanska et al., 2015) analyze the loss surface of a special random neural network through spin-glass theory and show that it has exponentially many local optima, whose loss is small and close to that of a global optimum. Later on, (Kawaguchi, 2016) eliminate some assumptions made by (Choromanska et al., 2015) but still require the independence of activations as (Choromanska et al., 2015), which is unrealistic. (Safran & Shamir, 2016) study the geometric structure of the neural network objective function. They have shown that with high probability random initialization will fall into a basin with a small objective value when the network is over-parameterized. (Livni et al., 2014) consider polynomial networks where the activations are square functions, which are typically not used in practice. (Haeffele & Vidal, 2015) show that when a local minimum has zero parameters related to a hidden node, a global optimum is achieved. (Freeman & Bruna, 2016) study the landscape of 1NN in terms of topology and geometry, and show that the level set becomes connected as the network is increasingly over-parameterized. (Hardt & Ma, 2017) show that products of matrices don't have spurious local minima and that deep residual networks can represent any function on a sample, as long as the number of parameters is larger than the sample size. (Soudry & Carmon, 2016) consider over-specified NNs, where the number of samples is smaller than the number of weights. (Dauphin et al., 2014) propose a new approach to second-order optimization that identifies and attacks the saddle point problem in high-dimensional non-convex optimization. They apply the approach to recurrent neural networks

and show practical performance. (Arora et al., 2017) use results from tropical geometry to show global optimality of an algorithm, but it requires $(2n)^k \operatorname{poly}(n)$ computational complexity.

Almost all of these results require the number of parameters is larger than the number of points, which probably overfits the model and no generalization performance will be guaranteed. In this paper, we propose an efficient and provable algorithm for 1NNs that can achieve the underlying ground-truth parameters.

2.3. Generalization Bound / Recovery Guarantees

The achievability of global optima of the objective from the training data doesn't guarantee the learned model to be able to generalize well on unseen testing data. In the literature, we find three main approaches to generalization guarantees.

1) *Use generalization analysis frameworks*, including VC dimension/Rademacher complexity, to bound the generalization error. A few works have studied the generalization performance for NNs. (Xie et al., 2017) follow (Soudry & Carmon, 2016) but additionally provide generalization bounds using Rademacher complexity. They assume the obtained parameters are in a regularization set so that the generalization performance is guaranteed, but this assumption can't be justified theoretically. (Hardt et al., 2016) apply stability analysis to the generalization analysis of SGD for convex and non-convex problems, arguing early stopping is important for generalization performance.

2) *Assume an underlying model and try to recover this model.* This direction is popular for many non-convex problems including matrix completion/sensing (Jain et al., 2013; Hardt, 2014; Sun & Luo, 2015; Balcan et al., 2017), mixed linear regression (Zhong et al., 2016), subspace recovery (Elhamifar & Vidal, 2009) and other latent models (Anandkumar et al., 2014).

Without making any assumptions, those non-convex problems are intractable (Arora et al., 2012a; Gillis & Vavasis, 2015; Song et al., 2017a; Gillis & Glineur, 2011; Razenshteyn et al., 2016; Sontag & Roy, 2011; Hardt & Moitra, 2013; Arora et al., 2012b; Yi et al., 2014). Recovery guarantees for NNs also need assumptions. Several different approaches under different assumptions are provided to have recovery guarantees on different NN settings.

Tensor methods (Anandkumar et al., 2014; Wang et al., 2015; Wang & Anandkumar, 2016; Song et al., 2016) are a general tool for recovering models with latent factors by assuming the data distribution is known. Some existing recovery guarantees for NNs are provided by tensor methods (Sedghi & Anandkumar, 2015; Janzamin et al., 2015). However, (Sedghi & Anandkumar, 2015) only provide guarantees to recover the subspace spanned by the weight matrix and no sample complexity is given, while (Janzamin et al., 2015) require $O(d^3/\epsilon^2)$ sample complexity. In this paper, we use tensor methods as an initialization step so that we don't need very accurate estimation of the moments, which enables us to reduce the total sample complexity from $1/\epsilon^2$ to $\log(1/\epsilon)$.

(Arora et al., 2014) provide polynomial sample complexity and computational complexity bounds for learning deep representations in unsupervised setting, and they need to assume the weights are sparse and randomly distributed in $[-1, 1]$.

(Tian, 2017) analyze 1NN by assuming Gaussian inputs in a supervised setting, in particular, regression and classification with a teacher. This paper also considers this setting. However, there are some key differences. a) (Tian, 2017) require the second-layer parameters are all ones, while we can learn these parameters. b) In (Tian, 2017), the ground-truth first-layer weight vectors are required to be orthogonal, while we only require linear independence. c) (Tian, 2017) require a good initialization but doesn't provide initialization methods, while we show the parameters can be efficiently initialized by tensor methods. d) In (Tian, 2017), only the population case (infinite sample size) is considered, so there is no sample complexity analysis, while we show finite sample complexity.

Recovery guarantees for convolution neural network with Gaussian inputs are provided in (Brutzkus & Globerson, 2017), where they show a globally converging guarantee of gradient descent on a one-hidden-layer no-overlap convolution neural network. However, they consider population case, so no sample complexity is provided. Also their analysis depends on ReLU activations and the no-overlap case is very unlikely to be used in practice. In this paper, we consider a large range of activation functions, but for one-hidden-layer fully-connected NNs.

3) *Improper Learning.* In the improper learning setting for NNs, the learning algorithm is not restricted to output a NN, but only should output a prediction function whose error is not much larger than the error of the best NN among all the NNs considered. (Zhang et al., 2016a;b) propose kernel methods to learn the prediction function which is guaranteed to have generalization performance close to that of the NN. However, the sample complexity and computational complexity are exponential. (Aslan et al., 2014) transform NNs to convex semi-definite programming. The works by (Bach, 2014) and (Bengio et al., 2005) are also in this direction. However, these methods are actually not learning the original NNs. Another work by (Zhang et al., 2017b) uses random initializations to achieve arbitrary small excess risk. However, their algorithm has exponential running time in $1/\epsilon$.

Roadmap. The paper is organized as follows. In Section 3, we present our problem setting and show three key properties of activations required for our guarantees. In Section 4, we introduce the formal theorem of local strong convexity and show local linear convergence for smooth activations. Section 5 presents a tensor method to initialize the parameters so that they fall into the basin of the local strong convexity region.

2.4. Notation

For any positive integer n, we use $[n]$ to denote the set $\{1, 2, \cdots, n\}$. For random variable X, let $\mathbb{E}[X]$ denote the expectation of X (if this quantity exists). For any vector $x \in \mathbb{R}^n$, we use $\|x\|$ to denote its ℓ_2 norm. We provide several definitions related to matrix A. Let $\det(A)$ denote the determinant of a square matrix A. Let $A^\top$ denote the transpose of A. Let $A^\dagger$ denote the Moore-Penrose pseudoinverse of A. Let A^{-1} denote the inverse of a full rank square matrix. Let $\|A\|_F$ denote the Frobenius norm of matrix A. Let $\|A\|$ denote the spectral norm of matrix A. Let $\sigma_i(A)$ to denote the i-th largest singular value of A. For any function f, we define $\widetilde{O}(f)$ to be $f \cdot \log^{O(1)}(f)$. In addition to $O(\cdot)$ notation, for two functions f, g, we use the shorthand $f \lesssim g$ (resp. $\gtrsim$) to indicate that $f \leq Cg$ (resp. $\geq$) for an absolute constant C. We use $\otimes$ to denote outer product and $\cdot$ to denote dot product. Given two column vectors $u, v \in \mathbb{R}^n$, then $u \otimes v \in \mathbb{R}^{n \times n}$ and $(u \otimes v)_{i,j} = u_i \cdot v_j$, and $u^\top v = \sum_{i=1}^n u_i v_i \in \mathbb{R}$. Given three column vectors $u, v, w \in \mathbb{R}^n$, then $u \otimes v \otimes w \in \mathbb{R}^{n \times n \times n}$ and $(u \otimes v \otimes w)_{i,j,k} = u_i \cdot v_j \cdot w_k$. We use $u^{\otimes r} \in \mathbb{R}^{n^r}$ to denote the vector u's outer product with itself $r-1$ times.

3. Problem Formulation

We consider the following regression problem. Given a set of n samples

$$S = \{(x_1, y_1), (x_2, y_2), \cdots (x_n, y_n)\} \subset \mathbb{R}^d \times \mathbb{R},$$

let $\mathcal{D}$ denote a underlying distribution over $\mathbb{R}^d \times \mathbb{R}$ with parameters

$$\{w_1^*, w_2^*, \cdots w_k^*\} \subset \mathbb{R}^d, \text{ and } \{v_1^*, v_2^*, \cdots, v_k^*\} \subset \mathbb{R}$$

such that each sample $(x, y) \in S$ is sampled i.i.d. from this distribution, with

$$\mathcal{D}: \qquad x \sim \mathcal{N}(0, I), \;\; y = \sum_{i=1}^k v_i^* \cdot \phi(w_i^{*\top} x), \qquad (1)$$

where $\phi(z)$ is the activation function, k is the number of nodes in the hidden layer. The main question we want to answer is: How many samples are sufficient to recover the underlying parameters?

It is well-known that, training one hidden layer neural network is NP-complete (Blum & Rivest, 1988). Thus, without making any assumptions, learning deep neural network is intractable. Throughout the paper, we assume x follows a standard normal distribution; the data is noiseless; the dimension of input data is at least the number of hidden nodes; and activation function $\phi(z)$ satisfies some reasonable properties.

Actually our results can be easily extended to multivariate Gaussian distribution with positive definite covariance and zero mean since we can estimate the covariance first and then transform the input to a standard normal distribution but with some loss of accuracy. Although this paper focuses on the regression problem, we can transform classification problems to regression problems if a good teacher is provided as described in (Tian, 2017). Our analysis requires k to be no greater than d, since the first-layer parameters will be linearly dependent otherwise.

For activation function $\phi(z)$, we assume it is continuous and if it is non-smooth let its first derivative be left derivative. Furthermore, we assume it satisfies Property 3.1, 3.2, and 3.3. These properties are critical for the later analyses. We also observe that most activation functions actually satisfy these three properties.

Property 3.1. *The first derivative $\phi'(z)$ is nonnegative and homogeneously bounded, i.e., $0 \leq \phi'(z) \leq L_1|z|^p$ for some constants $L_1 > 0$ and $p \geq 0$.*

Property 3.2. *Let $\alpha_q(\sigma) = \mathbb{E}_{z\sim\mathcal{N}(0,1)}[\phi'(\sigma \cdot z)z^q], \forall q \in \{0, 1, 2\}$, and $\beta_q(\sigma) = \mathbb{E}_{z\sim\mathcal{N}(0,1)}[\phi'^2(\sigma \cdot z)z^q], \forall q \in \{0, 2\}$. Let $\rho(\sigma)$ denote $\min\{\beta_0(\sigma) - \alpha_0^2(\sigma) - \alpha_1^2(\sigma),\ \beta_2(\sigma) - \alpha_1^2(\sigma) - \alpha_2^2(\sigma),\ \alpha_0(\sigma) \cdot \alpha_2(\sigma) - \alpha_1^2(\sigma)\}$ The first derivative $\phi'(z)$ satisfies that, for all $\sigma > 0$, we have $\rho(\sigma) > 0$.*

Property 3.3. *The second derivative $\phi''(z)$ is either* **(a)** *globally bounded $|\phi''(z)| \leq L_2$ for some constant L_2, i.e., $\phi(z)$ is L_2-smooth, or* **(b)** *$\phi''(z) = 0$ except for e (e is a finite constant) points.*

Remark 3.4. *The first two properties are related to the first derivative $\phi'(z)$ and the last one is about the second derivative $\phi''(z)$. At high level, Property 3.1 requires ϕ to be non-decreasing with homogeneously bounded derivative; Property 3.2 requires ϕ to be highly non-linear; Property 3.3 requires ϕ to be either smooth or piece-wise linear.*

Theorem 3.5. ReLU $\phi(z) = \max\{z, 0\}$, *leaky* ReLU $\phi(z) = \max\{z, 0.01z\}$, *squared* ReLU $\phi(z) = \max\{z, 0\}^2$ *and any non-linear non-decreasing smooth functions with bounded symmetric $\phi'(z)$, like the sigmoid function $\phi(z) = 1/(1+e^{-z})$, the* $\tanh$ *function and the* erf *function $\phi(z) = \int_0^z e^{-t^2} dt$, satisfy Property 3.1,3.2,3.3. The linear function, $\phi(z) = z$, doesn't satisfy Property 3.2 and the quadratic function, $\phi(z) = z^2$, doesn't satisfy Property 3.1 and 3.2.*

The proof can be found in the full version (Zhong et al., 2017).

4. Positive Definiteness of Hessian

In this section, we study the Hessian of empirical risk near the ground truth. We consider the case when v^* is already known. Note that for homogeneous activations, we can assume $v_i^* \in \{-1, 1\}$ since $v\phi(z) = \frac{v}{|v|}\phi(|v|^{1/p}z)$, where p is the degree of homogeneity. As v_i^* only takes discrete values for homogeneous activations, in the next section, we show we can exactly recover v^* using tensor methods with finite samples.

For a set of samples S, we define the *Empirical Risk*,

$$\widehat{f}_S(W) = \frac{1}{2|S|}\sum_{(x,y)\in S}\left(\sum_{i=1}^{k} v_i^*\phi(w_i^\top x) - y\right)^2. \quad (2)$$

For a distribution $\mathcal{D}$, we define the *Expected Risk*,

$$f_{\mathcal{D}}(W) = \frac{1}{2}\mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[\left(\sum_{i=1}^{k} v_i^*\phi(w_i^\top x) - y\right)^2\right]. \quad (3)$$

Let's calculate the gradient and the Hessian of $\widehat{f}_S(W)$ and $f_{\mathcal{D}}(W)$. For each $j \in [k]$, the partial gradient of $f_{\mathcal{D}}(W)$ with respect to w_j can be represented as

$$\frac{\partial f_{\mathcal{D}}(W)}{\partial w_j} = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[\left(\sum_{i=1}^{k} v_i^*\phi(w_i^\top x) - y\right) v_j^*\phi'(w_j^\top x)x\right].$$

For each $j, l \in [k]$ and $j \neq l$, the second partial derivative of $f_{\mathcal{D}}(W)$ for the (j,l)-th off-diagonal block is,

$$\frac{\partial^2 f_{\mathcal{D}}(W)}{\partial w_j \partial w_l} = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[v_j^* v_l^*\phi'(w_j^\top x)\phi'(w_l^\top x)xx^\top\right],$$

and for each $j \in [k]$, the second partial derivative of $f_{\mathcal{D}}(W)$ for the j-th diagonal block is

$$\frac{\partial^2 f_{\mathcal{D}}(W)}{\partial w_j^2} = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}[(\sum_{i=1}^{k} v_i^*\phi(w_i^\top x) - y)v_j^*\phi''(w_j^\top x)xx^\top + (v_j^*\phi'(w_j^\top x))^2 xx^\top].$$

If $\phi(z)$ is non-smooth, we use the Dirac function and its derivatives to represent $\phi''(z)$. Replacing the expectation $\mathbb{E}_{(x,y)\sim\mathcal{D}}$ by the average over the samples $|S|^{-1}\sum_{(x,y)\in S}$, we obtain the Hessian of the empirical risk.

Considering the case when $W = W^* \in \mathbb{R}^{d\times k}$, for all $j, l \in [k]$, we have,

$$\frac{\partial^2 f_{\mathcal{D}}(W^*)}{\partial w_j \partial w_l} = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[v_j^* v_l^*\phi'(w_j^{*\top} x)\phi'(w_l^{*\top} x)xx^\top\right].$$

If Property 3.3(b) is satisfied, $\phi''(z) = 0$ almost surely. So in this case the diagonal blocks of the empirical Hessian can be written as,

$$\frac{\partial^2 \widehat{f}_S(W)}{\partial w_j^2} = \frac{1}{|S|}\sum_{(x,y)\in S}(v_j^*\phi'(w_j^\top x))^2 xx^\top.$$

Now we show the Hessian of the objective near the global optimum is positive definite.

Definition 4.1. *Given the ground truth matrix $W^* \in \mathbb{R}^{d\times k}$, let $\sigma_i(W^*)$ denote the i-th singular value of W^*, often abbreviated as σ_i. Let $\kappa = \sigma_1/\sigma_k$, $\lambda = (\prod_{i=1}^{k}\sigma_i)/\sigma_k^k$. Let $v_{\max}$ denote $\max_{i\in[k]}|v_i^*|$ and $v_{\min}$ denote $\min_{i\in[k]}|v_i^*|$. Let $\nu = v_{\max}/v_{\min}$. Let ρ denote $\rho(\sigma_k)$. Let $\tau = (3\sigma_1/2)^{4p}/\min_{\sigma\in[\sigma_k/2, 3\sigma_1/2]}\{\rho^2(\sigma)\}$.*

Theorem 4.2. *For any $W \in \mathbb{R}^{d\times k}$ with $\|W - W^*\| \leq \mathrm{poly}(1/k, 1/\lambda, 1/\nu, \rho/\sigma_1^{2p})\cdot\|W^*\|$, let S denote a set of i.i.d. samples from distribution $\mathcal{D}$ (defined in (1)) and let the activation function satisfy Property 3.1,3.2,3.3. Then for any $t \geq 1$, if $|S| \geq d\cdot\mathrm{poly}(\log d, t, k, \nu, \tau, \lambda, \sigma_1^{2p}/\rho)$, we have with probability at least $1 - d^{-\Omega(t)}$,*

$$\Omega(v_{\min}^2\rho(\sigma_k)/(\kappa^2\lambda))I \preceq \nabla^2\widehat{f}_S(W) \preceq O(kv_{\max}^2\sigma_1^{2p})I.$$

Remark 4.3. *As we can see from Theorem 4.2, $\rho(\sigma_k)$ from Property 3.2 plays an important role for positive definite (PD) property. Interestingly, many popular activations, like ReLU, sigmoid and tanh, have $\rho(\sigma_k) > 0$, while some simple functions like linear ($\phi(z) = z$) and square ($\phi(z) = z^2$) functions have $\rho(\sigma_k) = 0$ and their Hessians are rank-deficient. Another important numbers are κ and λ, two different condition numbers of the weight matrix, which directly influences the positive definiteness. If W^* is rank deficient, $\lambda \to \infty$, $\kappa \to \infty$ and we don't have PD property. In the best case when W^* is orthogonal, $\lambda = \kappa = 1$. In the worse case, λ can be exponential in k. Also W should be close enough to W^*. In the next section, we provide tensor methods to initialize w_i^* and v_i^* such that they satisfy the conditions in Theorem 4.2.*

For the PD property to hold, we need the samples to be independent of the current parameters. Therefore, we need to do resampling at each iteration to guarantee the convergence in iterative algorithms like gradient descent. The following theorem provides the linear convergence guarantee of gradient descent for smooth activations.

Theorem 4.4 (Linear convergence of gradient descent)**.** *Let W be the current iterate satisfying $\|W - W^*\| \leq \mathrm{poly}(1/\nu, 1/k, 1/\lambda, \rho/\sigma_1^{2p})\|W^*\|$. Let S denote a set of i.i.d. samples from distribution $\mathcal{D}$ (defined in (1)) with $|S| \geq d\cdot\mathrm{poly}(\log d, t, k, \nu, \tau, \lambda, \sigma_1^{2p}/\rho)$ and let the activation function satisfy Property 3.1,3.2 and 3.3(a). Define $m_0 := \Theta(v_{\min}^2\rho(\sigma_k)/(\kappa^2\lambda))$ and $M_0 := \Theta(kv_{\max}^2\sigma_1^{2p})$.*

If we perform gradient descent with step size $1/M_0$ *on* $\widehat{f}_S(W)$ *and obtain the next iterate,*

$$\widetilde{W} = W - \frac{1}{M_0}\nabla \widehat{f}_S(W),$$

then with probability at least $1 - d^{-\Omega(t)}$*,*

$$\|\widetilde{W} - W^*\|_F^2 \leq (1 - \frac{m_0}{M_0})\|W - W^*\|_F^2.$$

Due to the space limitation, we provide the proofs in the full version.

5. Tensor Methods for Initialization

In this section, we show that Tensor methods can recover the parameters W^* to some precision and exactly recover v^* for homogeneous activations.

It is known that most tensor problems are NP-hard (Håstad, 1990; Hillar & Lim, 2013) or even hard to approximate (Song et al., 2017b). However, by making some assumptions, tensor decomposition method becomes efficient (Anandkumar et al., 2014; Wang et al., 2015; Wang & Anandkumar, 2016; Song et al., 2016). Here we utilize the noiseless assumption and Gaussian inputs assumption to show a provable and efficient tensor methods.

5.1. Preliminary

Let's define a special outer product $\widetilde{\otimes}$ for simplification of the notation. If $v \in \mathbb{R}^d$ is a vector and I is the identity matrix, then $v\widetilde{\otimes}I = \sum_{j=1}^d [v \otimes e_j \otimes e_j + e_j \otimes v \otimes e_j + e_j \otimes e_j \otimes v]$. If M is a symmetric rank-r matrix factorized as $M = \sum_{i=1}^r s_i v_i v_i^\top$ and I is the identity matrix, then

$$M\widetilde{\otimes}I = \sum_{i=1}^r s_i \sum_{j=1}^d \sum_{l=1}^6 A_{l,i,j},$$

where $A_{1,i,j} = v_i \otimes v_i \otimes e_j \otimes e_j$, $A_{2,i,j} = v_i \otimes e_j \otimes v_i \otimes e_j$, $A_{3,i,j} = e_j \otimes v_i \otimes v_i \otimes e_j$, $A_{4,i,j} = v_i \otimes e_j \otimes e_j \otimes v_i$, $A_{5,i,j} = e_j \otimes v_i \otimes e_j \otimes v_i$ and $A_{6,i,j} = e_j \otimes e_j \otimes v_i \otimes v_i$.

Denote $\overline{w} = w/\|w\|$. Now let's calculate some moments.

Definition 5.1. *We define* M_1, M_2, M_3, M_4 *and* $m_{1,i}, m_{2,i}, m_{3,i}, m_{4,i}$ *as follows :*
$M_1 = \mathbb{E}_{(x,y)\sim\mathcal{D}}[y \cdot x]$.
$M_2 = \mathbb{E}_{(x,y)\sim\mathcal{D}}[y \cdot (x \otimes x - I)]$.
$M_3 = \mathbb{E}_{(x,y)\sim\mathcal{D}}[y \cdot (x^{\otimes 3} - x\widetilde{\otimes}I)]$.
$M_4 = \mathbb{E}_{(x,y)\sim\mathcal{D}}[y \cdot (x^{\otimes 4} - (x \otimes x)\widetilde{\otimes}I + I\widetilde{\otimes}I)]$.
$\gamma_j(\sigma) = \mathbb{E}_{z\sim\mathcal{N}(0,1)}[\phi(\sigma \cdot z)z^j],\ \forall j = 0, 1, 2, 3, 4.$
$m_{1,i} = \gamma_1(\|w_i^*\|)$.
$m_{2,i} = \gamma_2(\|w_i^*\|) - \gamma_0(\|w_i^*\|)$.
$m_{3,i} = \gamma_3(\|w_i^*\|) - 3\gamma_1(\|w_i^*\|)$.
$m_{4,i} = \gamma_4(\|w_i^*\|) + 3\gamma_0(\|w_i^*\|) - 6\gamma_2(\|w_i^*\|)$.

According to Definition 5.1, we have the following results,

Claim 5.2. *For each* $j \in [4]$, $M_j = \sum_{i=1}^k v_i^* m_{j,i} \overline{w}_i^{*\otimes j}$.

Note that some $m_{j,i}$'s will be zero for specific activations. For example, for activations with symmetric first derivatives, i.e., $\phi'(z) = \phi'(-z)$, like sigmoid and erf, we have $\phi(z) + \phi(-z)$ being a constant and $M_2 = 0$ since $\gamma_0(\sigma) = \gamma_2(\sigma)$. Another example is ReLU. ReLU functions have vanishing M_3, i.e., $M_3 = 0$, as $\gamma_3(\sigma) = 3\gamma_1(\sigma)$. To make tensor methods work, we make the following assumption.

Assumption 5.3. *Assume the activation function* $\phi(z)$ *satisfies the following conditions:*
1. If $M_j \neq 0$*, then* $m_{j,i} \neq 0$ *for all* $i \in [k]$.
2. At least one of M_3 *and* M_4 *is non-zero.*
3. If $M_1 = M_3 = 0$*, then* $\phi(z)$ *is an even function, i.e.,* $\phi(z) = \phi(-z)$.
4. If $M_2 = M_4 = 0$*, then* $\phi(z)$ *is an odd function, i.e.,* $\phi(z) = -\phi(-z)$.

If $\phi(z)$ is an odd function then $\phi(z) = -\phi(-z)$ and $v\phi(w^\top x) = -v\phi(-w^\top x)$. Hence we can always assume $v > 0$. If $\phi(z)$ is an even function, then $v\phi(w^\top x) = v\phi(-w^\top x)$. So if w recovers w^* then $-w$ also recovers w^*. Note that ReLU, leaky ReLU and squared ReLU satisfy Assumption 5.3. We further define the following non-zero moments.

Definition 5.4. *Let* $\alpha \in \mathbb{R}^d$ *denote a randomly picked vector. We define* P_2 *and* P_3 *as follows:* $P_2 = M_{j_2}(I, I, \alpha, \cdots, \alpha)$ *, where* $j_2 = \min\{j \geq 2 | M_j \neq 0\}$ *and* $P_3 = M_{j_3}(I, I, I, \alpha, \cdots, \alpha)$*, where* $j_3 = \min\{j \geq 3 | M_j \neq 0\}$.

According to Definition 5.1 and 5.4, we have,

Claim 5.5. $P_2 = \sum_{i=1}^k v_i^* m_{j_2,i} (\alpha^\top \overline{w}_i^*)^{j_2-2} \overline{w}_i^{*\otimes 2}$ *and* $P_3 = \sum_{i=1}^k v_i^* m_{j_3,i} (\alpha^\top \overline{w}_i^*)^{j_3-3} \overline{w}_i^{*\otimes 3}$.

In other words for the above definition, P_2 is equal to the first non-zero matrix in the ordered sequence $\{M_2, M_3(I, I, \alpha), M_4(I, I, \alpha, \alpha)\}$. P_3 is equal to the first non-zero tensor in the ordered sequence $\{M_3, M_4(I, I, I, \alpha)\}$. Since α is randomly picked up, $w_i^{*\top}\alpha \neq 0$ and we view this number as a constant throughout this paper. So by construction and Assumption 5.3, both P_2 and P_3 are rank-k. Also, let $\widehat{P}_2 \in \mathbb{R}^{d\times d}$ and $\widehat{P}_3 \in \mathbb{R}^{d\times d\times d}$ denote the corresponding empirical moments of $P_2 \in \mathbb{R}^{d\times d}$ and $P_3 \in \mathbb{R}^{d\times d\times d}$ respectively.

5.2. Algorithm

Now we briefly introduce how to use a set of samples with size linear in dimension to recover the ground truth parameters to some precision. As shown in the previous section, we have a rank-k 3rd-order moment P_3 that

Algorithm 1 Initialization via Tensor Method

procedure INITIALIZATION(S) ▷ Theorem 5.6
  $S_2, S_3, S_4 \leftarrow$ PARTITION($S, 3$)
  $\widehat{P}_2 \leftarrow \mathbb{E}_{S_2}[P_2]$
  $V \leftarrow$ POWERMETHOD($\widehat{P}_2, k$)
  $\widehat{R}_3 \leftarrow \mathbb{E}_{S_3}[P_3(V, V, V)]$
  $\{\widehat{u}_i\}_{i\in[k]} \leftarrow$ KCL($\widehat{R}_3$)
  $\{w_i^{(0)}, v_i^{(0)}\}_{i\in[k]} \leftarrow$ RECMAGSIGN($V, \{\widehat{u}_i\}_{i\in[k]}, S_4$)
  Return $\{w_i^{(0)}, v_i^{(0)}\}_{i\in[k]}$
end procedure

has tensor decomposition formed by $\{\overline{w}_1^*, \overline{w}_2^*, \cdots, \overline{w}_k^*\}$. Therefore, we can use the non-orthogonal decomposition method (Kuleshov et al., 2015) to decompose the corresponding estimated tensor $\widehat{P}_3$ and obtain an approximation of the parameters. The precision of the obtained parameters depends on the estimation error of P_3, which requires $\Omega(d^3/\epsilon^2)$ samples to achieve ϵ error. Also, the time complexity for tensor decomposition on a $d \times d \times d$ tensor is $\Omega(d^3)$.

In this paper, we reduce the cubic dependency of sample/computational complexity in dimension (Janzamin et al., 2015) to linear dependency. Our idea follows the techniques used in (Zhong et al., 2016), where they first used a 2nd-order moment P_2 to approximate the subspace spanned by $\{\overline{w}_1^*, \overline{w}_2^*, \cdots, \overline{w}_k^*\}$, denoted as V, then use V to reduce a higher-dimensional third-order tensor $P_3 \in \mathbb{R}^{d\times d\times d}$ to a lower-dimensional tensor $R_3 := P_3(V, V, V) \in \mathbb{R}^{k\times k\times k}$. Since the tensor decomposition and the tensor estimation are conducted on a lower-dimensional $\mathbb{R}^{k\times k\times k}$ space, the sample complexity and computational complexity are reduced.

The detailed algorithm is shown in Algorithm 1. First, we randomly partition the dataset into three subsets each with size $\widetilde{O}(d)$. Then apply the power method on $\widehat{P}_2$, which is the estimation of P_2 from S_2, to estimate V. After that, the non-orthogonal tensor decomposition (KCL)(Kuleshov et al., 2015) on $\widehat{R}_3$ outputs $\widehat{u}_i$ which estimates $s_i V^\top \overline{w}_i^*$ for $i \in [k]$ with unknown sign $s_i \in \{-1, 1\}$. Hence $\overline{w}_i^*$ can be estimated by $s_i V \widehat{u}_i$. Finally we estimate the magnitude of w_i^* and the signs s_i, v_i^* in the RECMAGSIGN function for homogeneous activations. We discuss the details of each procedure and provide POWERMETHOD and RECMAGSIGN algorithms in the full version.

5.3. Theoretical Analysis

We formally present our theorem for Algorithm 1, and provide the proof in the full version.

Theorem 5.6. *Let the activation function be homogeneous satisfying Assumption 5.3. For any $0 < \epsilon < 1$ and $t \geq 1$, if $|S| \geq \epsilon^{-2} \cdot d \cdot \mathrm{poly}(t, k, \kappa, \log d)$, then there exists an algorithm (Algorithm 1) that takes $|S|k \cdot \widetilde{O}(d)$ time and outputs a matrix $W^{(0)} \in \mathbb{R}^{d\times k}$ and a vector $v^{(0)} \in \mathbb{R}^k$ such that, with probability at least $1 - d^{-\Omega(t)}$,*

$$\|W^{(0)} - W^*\|_F \leq \epsilon \cdot \mathrm{poly}(k, \kappa)\|W^*\|_F, \text{ and } v_i^{(0)} = v_i^*.$$

Algorithm 2 Globally Converging Algorithm

procedure LEARNING1NN(S, d, k, ϵ) ▷ Theorem 6.1
  $T \leftarrow \log(1/\epsilon) \cdot \mathrm{poly}(k, \nu, \lambda, \sigma_1^{2p}/\rho)$.
  $\eta \leftarrow 1/(kv_{\max}^2\sigma_1^{2p})$.
  $S_0, S_1, \cdots, S_q \leftarrow$ PARTITION($S, q+1$).
  $W^{(0)}, v^{(0)} \leftarrow$ INITIALIZATION(S_0).
  Set $v_i^* \leftarrow v_i^{(0)}$ in Eq. (2) for all $\widehat{f}_{S_q}(W)$, $q \in [T]$
  for $q = 0, 1, 2, \cdots, T-1$ **do**
    $W^{(q+1)} = W^{(q)} - \eta\nabla\widehat{f}_{S_{q+1}}(W^{(q)})$
  end for
  Return $\{w_i^{(T)}, v_i^{(0)}\}_{i\in[k]}$
end procedure

6. Global Convergence

Combining the positive definiteness of the Hessian near the global optimal in Section 4 and the tensor initialization methods in Section 5, we come up with the overall globally converging algorithm Algorithm 2 and its guarantee Theorem 6.1.

Theorem 6.1 (Global convergence guarantees)**.** *Let S denote a set of i.i.d. samples from distribution $\mathcal{D}$ (defined in (1)) and let the activation function be homogeneous satisfying Property 3.1, 3.2, 3.3(a) and Assumption 5.3. Then for any $t \geq 1$ and any $\epsilon > 0$, if $|S| \geq d\log(1/\epsilon) \cdot \mathrm{poly}(\log d, t, k, \lambda)$, $T \geq \log(1/\epsilon) \cdot \mathrm{poly}(k, \nu, \lambda, \sigma_1^{2p}/\rho)$ and $0 < \eta \leq 1/(kv_{\max}^2\sigma_1^{2p})$, then there is an Algorithm (procedure* LEARNING1NN *in Algorithm 2) taking $|S| \cdot d \cdot \mathrm{poly}(\log d, k, \lambda)$ time and outputting a matrix $W^{(T)} \in \mathbb{R}^{d\times k}$ and a vector $v^{(0)} \in \mathbb{R}^k$ satisfying*

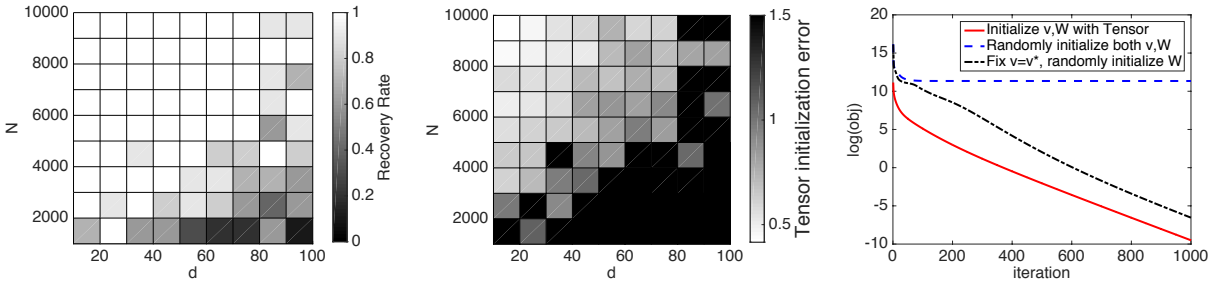
$$\|W^{(T)} - W^*\|_F \leq \epsilon\|W^*\|_F, \text{ and } v_i^{(0)} = v_i^*.$$

with probability at least $1 - d^{-\Omega(t)}$.

This follows by combining Theorem 4.4 and Theorem 5.6.

7. Numerical Experiments

In this section we use synthetic data to verify our theoretical results. We generate data points $\{x_i, y_i\}_{i=1,2,\cdots,n}$ from Distribution $\mathcal{D}$(defined in Eq. (1)). We set $W^* = U\Sigma V^\top$, where $U \in \mathbb{R}^{d\times k}$ and $V \in \mathbb{R}^{k\times k}$ are orthogonal matrices generated from QR decomposition of Gaussian matrices, Σ is a diagonal matrix whose diagonal elements



(a) Sample complexity for recovery (b) Tensor initialization error (c) Objective v.s. iterations

Figure 1. Numerical Experiments

are $1, 1+\frac{\kappa-1}{k-1}, 1+\frac{2(\kappa-1)}{k-1}, \cdots, \kappa$. In this experiment, we set $\kappa = 2$ and $k = 5$. We set v_i^* to be randomly picked from $\{-1, 1\}$ with equal chance. We use squared ReLU $\phi(z) = \max\{z, 0\}^2$, which is a smooth homogeneous function. For non-orthogonal tensor methods, we directly use the code provided by (Kuleshov et al., 2015) with the number of random projections fixed as $L = 100$. We pick the stepsize $\eta = 0.02$ for gradient descent. In the experiments, we don't do the resampling since the algorithm still works well without resampling.

First we show the number of samples required to recover the parameters for different dimensions. We fix $k = 5$, change d for $d = 10, 20, \cdots, 100$ and n for $n = 1000, 2000, \cdots, 10000$. For each pair of d and n, we run 10 trials. We say a trial successfully recovers the parameters if there exists a permutation $\pi : [k] \to [k]$, such that the returned parameters W and v satisfy

$$\max_{j\in[k]}\{\|w_j^* - w_{\pi(j)}\|/\|w_j^*\|\} \le 0.01 \text{ and } v_{\pi(j)} = v_j^*.$$

We record the recovery rates and represent them as grey scale in Fig. 1(a). As we can see from Fig. 1(a), the least number of samples required to have 100% recovery rate is about proportional to the dimension.

Next we test the tensor initialization. We show the error between the output of the tensor method and the ground truth parameters against the number of samples under different dimensions in Fig 1(b). The pure dark blocks indicate, in at least one of the 10 trials, $\sum_{i=1}^k v_i^{(0)} \neq \sum_{i=1}^k v_i^*$, which means $v_i^{(0)}$ is not correctly initialized. Let $\Pi(k)$ denote the set of all possible permutations $\pi : [k] \to [k]$. The grey scale represents the averaged error,

$$\min_{\pi\in\Pi(k)} \max_{j\in[k]}\{\|w_j^* - w_{\pi(j)}^{(0)}\|/\|w_j^*\|\},$$

over 10 trials. As we can see, with a fixed dimension, the more samples we have the better initialization we obtain. We can also see that to achieve the same initialization error, the sample complexity required is about proportional to the dimension.

We also compare different initialization methods for gradient descent in Fig. 1(c). We fix $d = 10, k = 5, n = 10000$ and compare three different initialization approaches, (I) Let both v and W be initialized from tensor methods, and then do gradient descent for W while v is fixed; (II) Let both v and W be initialized from random Gaussian, and then do gradient descent for both W and v; (III) Let $v = v^*$ and W be initialized from random Gaussian, and then do gradient descent for W while v is fixed. As we can see from Fig 1(c), Approach (I) is the fastest and Approach (II) doesn't converge even if more iterations are allowed. Both Approach (I) and (III) have linear convergence rate when the objective value is small enough, which verifies our local linear convergence claim.

8. Conclusion

As shown in Theorem 6.1, the tensor initialization followed by gradient descent will provide a globally converging algorithm with linear time/sample complexity in dimension, logarithmic in precision and polynomial in other factors for smooth homogeneous activation functions. Our distilled properties for activation functions include a wide range of non-linear functions and hopefully provide an intuition to understand the role of non-linear activations played in optimization. Deeper neural networks and convergence for SGD will be considered in the future.

Acknowledgments

P. L. Bartlett would like to gratefully acknowledge the support of Australian Research Council through an Australian Laureate Fellowship (FL110100281) and through the Australian Research Council Centre of Excellence for Mathematical and Statistical Frontiers (ACEMS), and the support of the NSF through grant IIS-1619362. I. S. Dhillon would like to gratefully acknowledge the support of NSF grants CCF-1320746, IIS-1546452 and CCF-1564000. Part of the work was done while K. Zhong was interning in Microsoft Research, India. The authors would like to thank Surbhi Goel, Adam Klivans, Qi Lei, Eric Price, David P. Woodruff, Peilin Zhong, Hongyang Zhang and Jiong Zhang for useful discussions.

References

Anandkumar, Animashree, Ge, Rong, Hsu, Daniel, Kakade, Sham M, and Telgarsky, Matus. Tensor decompositions for learning latent variable models. *JMLR*, 15:2773–2832, 2014.

Andoni, Alexandr, Panigrahy, Rina, Valiant, Gregory, and Zhang, Li. Learning polynomials with neural networks. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, pp. 1908–1916, 2014.

Arora, Sanjeev, Ge, Rong, Kannan, Ravindran, and Moitra, Ankur. Computing a nonnegative matrix factorization–provably. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing (STOC)*, pp. 145–162. ACM, 2012a.

Arora, Sanjeev, Ge, Rong, and Moitra, Ankur. Learning topic models–going beyond svd. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pp. 1–10. IEEE, 2012b.

Arora, Sanjeev, Bhaskara, Aditya, Ge, Rong, and Ma, Tengyu. Provable bounds for learning some deep representations. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, pp. 584–592. https://arxiv.org/pdf/1310.6343.pdf, 2014.

Arora, Sanjeev, Ge, Rong, Ma, Tengyu, and Risteski, Andrej. Provable learning of noisy-or networks. In *Proceedings of the 49th Annual Symposium on the Theory of Computing (STOC)*. https://arxiv.org/pdf/1612.08795.pdf, 2017.

Aslan, Özlem, Zhang, Xinhua, and Schuurmans, Dale. Convex deep learning via normalized kernels. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 3275–3283, 2014.

Bach, Francis. Breaking the curse of dimensionality with convex neural networks. *arXiv preprint arXiv:1412.8690*, 2014.

Balcan, Maria-Florina, Liang, Yingyu, Woodruff, David P., and Zhang, Hongyang. Optimal sample complexity for matrix completion and related problems via ℓ_2-regularization. *arXiv preprint arXiv:1704.08683*, 2017.

Balduzzi, David. Deep online convex optimization with gated games. *arXiv preprint arXiv:1604.01952*, 2016.

Balduzzi, David, McWilliams, Brian, and Butler-Yeoman, Tony. Neural taylor approximations: Convergence and exploration in rectifier networks. *arXiv preprint arXiv:1611.02345*, 2016.

Bartlett, Peter L. The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. *IEEE Transactions on Information Theory*, 44(2):525–536, 1998.

Bengio, Yoshua, Roux, Nicolas L, Vincent, Pascal, Delalleau, Olivier, and Marcotte, Patrice. Convex neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 123–130, 2005.

Blum, Avrim and Rivest, Ronald L. Training a 3-node neural network is np-complete. In *Proceedings of the 1st International Conference on Neural Information Processing Systems (NIPS)*, pp. 494–501. MIT Press, 1988.

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

Choromanska, Anna, Henaff, MIkael, Mathieu, Michael, Ben Arous, Gerard, and LeCun, Yann. The loss surfaces of multilayer networks. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 192–204, 2015.

Cohen, Nadav and Shashua, Amnon. Convolutional rectifier networks as generalized tensor decompositions. In *International Conference on Machine Learning (ICML)*, 2016.

Cohen, Nadav, Sharir, Or, and Shashua, Amnon. On the expressive power of deep learning: A tensor analysis. In *29th Annual Conference on Learning Theory (COLT)*, pp. 698–728, 2016.

Daniely, Amit, Frostig, Roy, and Singer, Yoram. Toward deeper understanding of neural networks: The power of initialization and a dual view on expressivity. In *Advances in neural information processing systems (NIPS)*, pp. 2253–2261, 2016.

Dauphin, Yann N, Pascanu, Razvan, Gulcehre, Caglar, Cho, Kyunghyun, Ganguli, Surya, and Bengio, Yoshua. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in neural information processing systems (NIPS)*, pp. 2933–2941, 2014.

Elhamifar, Ehsan and Vidal, René. Sparse subspace clustering. In *CVPR*, pp. 2790–2797, 2009.

Feng, Jiashi, Zahavy, Tom, Kang, Bingyi, Xu, Huan, and Mannor, Shie. Ensemble robustness of deep learning algorithms. *arXiv preprint arXiv:1602.02389*, 2016.

Freeman, C Daniel and Bruna, Joan. Topology and geometry of half-rectified network optimization. In *arXiv preprint*. https://arxiv.org/pdf/1611.01540.pdf, 2016.

Gillis, Nicolas and Glineur, François. Low-rank matrix approximation with weights or missing data is np-hard. *SIAM Journal on Matrix Analysis and Applications*, 32(4):1149–1165, 2011.

Gillis, Nicolas and Vavasis, Stephen A. On the complexity of robust pca and ℓ_1-norm low-rank matrix approximation. *arXiv preprint arXiv:1509.09236*, 2015.

Goel, Surbhi, Kanade, Varun, Klivans, Adam, and Thaler, Justin. Reliably learning the relu in polynomial time. In *30th Annual Conference on Learning Theory (COLT)*. https://arxiv.org/pdf/1611.10258.pdf, 2017.

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

Hardt, Moritz. Understanding alternating minimization for matrix completion. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pp. 651–660. IEEE, 2014.

Hardt, Moritz and Ma, Tengyu. Identity matters in deep learning. *ICLR*, 2017.

Hardt, Moritz and Moitra, Ankur. Algorithms and hardness for robust subspace recovery. In *COLT*, volume 30, pp. 354–375, 2013.

Hardt, Moritz, Recht, Ben, and Singer, Yoram. Train faster, generalize better: Stability of stochastic gradient descent. In *ICML*, pp. 1225–1234, 2016.

Håstad, Johan. Tensor rank is np-complete. *Journal of Algorithms*, 11(4):644–654, 1990.

Hillar, Christopher J and Lim, Lek-Heng. Most tensor problems are np-hard. In *Journal of the ACM (JACM)*, volume 60(6), pp. 45. https://arxiv.org/pdf/0911.1393.pdf, 2013.

Hornik, Kurt. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

Jain, Prateek, Netrapalli, Praneeth, and Sanghavi, Sujay. Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing (STOC)*, 2013.

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

Kawaguchi, Kenji. Deep learning without poor local minima. *arXiv preprint arXiv:1605.07110*, 2016.

Kuleshov, Volodymyr, Chaganty, Arun, and Liang, Percy. Tensor factorization via matrix factorization. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 507–516, 2015.

Livni, Roi, Shalev-Shwartz, Shai, and Shamir, Ohad. On the computational efficiency of training neural networks. In *Advances in neural information processing systems (NIPS)*, pp. 855–863, 2014.

Montufar, Guido F, Pascanu, Razvan, Cho, Kyunghyun, and Bengio, Yoshua. On the number of linear regions of deep neural networks. In *Advances in neural information processing systems (NIPS)*, pp. 2924–2932, 2014.

Poole, Ben, Lahiri, Subhaneil, Raghu, Maithreyi, Sohl-Dickstein, Jascha, and Ganguli, Surya. Exponential expressivity in deep neural networks through transient chaos. In *Advances In Neural Information Processing Systems (NIPS)*, pp. 3360–3368, 2016.

Raghu, Maithra, Poole, Ben, Kleinberg, Jon, Ganguli, Surya, and Sohl-Dickstein, Jascha. On the expressive power of deep neural networks. *arXiv preprint arXiv:1606.05336*, 2016.

Razenshteyn, Ilya P, Song, Zhao, and Woodruff, David P. Weighted low rank approximations with provable guarantees. In *Proceedings of the 48th Annual Symposium on the Theory of Computing (STOC)*, pp. 250–263, 2016.

Safran, Itay and Shamir, Ohad. On the quality of the initial basin in overspecified neural networks. In *International Conference on Machine Learning (ICML)*, 2016.

Sagun, Levent, Bottou, Léon, and LeCun, Yann. Singularity of the Hessian in deep learning. *arXiv preprint arXiv:1611.07476*, 2016.

Sedghi, Hanie and Anandkumar, Anima. Provable methods for training neural networks with sparse connectivity. In *International Conference on Learning Representation (ICLR)*, 2015.

Shamir, Ohad. Distribution-specific hardness of learning neural networks. *arXiv preprint arXiv:1609.01037*, 2016.

Song, Zhao, Woodruff, David P., and Zhang, Huan. Sublinear time orthogonal tensor decomposition. In *Advances in Neural Information Processing Systems(NIPS)*, pp. 793–801, 2016.

Song, Zhao, Woodruff, David P., and Zhong, Peilin. Low rank approximation with entrywise ℓ_1-norm error. In *Proceedings of the 49th Annual Symposium on the Theory of Computing (STOC)*. ACM, `https://arxiv.org/pdf/1611.00898.pdf`, 2017a.

Song, Zhao, Woodruff, David P., and Zhong, Peilin. Relative error tensor low rank approximation. In *arXiv preprint*. `https://arxiv.org/pdf/1704.08246.pdf`, 2017b.

Sontag, David and Roy, Dan. Complexity of inference in latent dirichlet allocation. In *Advances in neural information processing systems*, pp. 1008–1016, 2011.

Soudry, Daniel and Carmon, Yair. No bad local minima: Data independent training error guarantees for multilayer neural networks. *arXiv preprint arXiv:1605.08361*, 2016.

Sun, Ruoyu and Luo, Zhi-Quan. Guaranteed matrix completion via non-convex factorization. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 270–289. IEEE, 2015.

Swirszcz, Grzegorz, Czarnecki, Wojciech Marian, and Pascanu, Razvan. Local minima in training of deep networks. *arXiv preprint arXiv:1611.06310*, 2016.

Telgarsky, Matus. Benefits of depth in neural networks. In *29th Annual Conference on Learning Theory (COLT)*, pp. 1517–1539, 2016.

Tian, Yuandong. Symmetry-breaking convergence analysis of certain two-layered neural networks with ReLU nonlinearity. In *Workshop at International Conference on Learning Representation*, 2017.

Wang, Yining and Anandkumar, Anima. Online and differentially-private tensor decomposition. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 3531–3539, 2016.

Wang, Yining, Tung, Hsiao-Yu, Smola, Alexander J, and Anandkumar, Anima. Fast and guaranteed tensor decomposition via sketching. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 991–999, 2015.

Xie, Bo, Liang, Yingyu, and Song, Le. Diversity leads to generalization in neural networks. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

Yi, Xinyang, Caramanis, Constantine, and Sanghavi, Sujay. Alternating minimization for mixed linear regression. In *ICML*, pp. 613–621, 2014.

Zhang, Chiyuan, Bengio, Samy, Hardt, Moritz, Recht, Benjamin, and Vinyals, Oriol. Understanding deep learning requires rethinking generalization. In *ICLR*, 2017a.

Zhang, Yuchen, Lee, Jason D, and Jordan, Michael I. L1-regularized neural networks are improperly learnable in polynomial time. In *Proceedings of The 33rd International Conference on Machine Learning (ICML)*, pp. 993–1001, 2016a.

Zhang, Yuchen, Liang, Percy, and Wainwright, Martin J. Convexified convolutional neural networks. *arXiv preprint arXiv:1609.01000*, 2016b.

Zhang, Yuchen, Lee, Jason D., Wainwright, Martin J., and Jordan, Michael I. On the learnability of fully-connected neural networks. In *International Conference on Artificial Intelligence and Statistics*, 2017b.

Zhong, Kai, Jain, Prateek, and Dhillon, Inderjit S. Mixed linear regression with multiple components. In *Advances in neural information processing systems (NIPS)*, pp. 2190–2198, 2016.

Zhong, Kai, Song, Zhao, Jain, Prateek, Bartlett, Peter L., and Dhillon, Inderjit S. Recovery guarantees for one-hidden-layer neural networks. In *ICML*. `https://arxiv.org/pdf/1706.03175.pdf`, 2017.