LEARNING SHALLOW NEURAL NETWORKS VIA PROVABLE GRADIENT DESCENT WITH RANDOM INITIALIZATION

Shuhao Xia and Yuanming Shi

School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China
Email: {xiashh, shiym}@shanghaitech.edu.cn

ABSTRACT

This paper presents the provable gradient descent algorithm with *random initialization* for learning a two-layer neural network with quadratic activation functions. Specifically, we focus on the under-parameterized regime where the number of hidden units is smaller than the dimension of the inputs. We reveal that the randomly initialized gradient descent for the nonconvex neural network training problem is able to enter a local region that enjoys strong convexity and strong smoothness within a few iterations, and then provably converges to a globally optimal model at a linear rate.

***Index Terms*—** Polynomial neural network, gradient descent, nonconvex optimization, local landscape, random initialization.

1. INTRODUCTION

Deep learning has recently emerged as a powerful tool in large scale machine learning systems. Various neural networks lead to great influence on diversified applications, such as computer vision, natural language processing and reinforcement learning [1]. However, despite the empirically successful performance of neural networks in practices, it is critical to understand the provable methods for learning neural networks. To achieve this goal, the main challenge is how to deal with high-dimensional and nonconvex statistical optimization problems arising from training neural networks. There is a growing body of recent works to tame the nonconvexity in solving the nonconvex optimization problems in deep neural networks. Although the nuclear norm relaxation is able to provide performance guarantees for convolutional neural networks in [2], the convex approaches are computationally expensive to deal with large-scale data set.

The global landscape analysis for the loss functions becomes a powerful tool to tame nonconvexity. Specifically, with enough training data, some nonconvex loss functions enjoy benign geometric structures that all the local minima are as good as global minima, and all the saddle points can be escaped. In particular, the loss functions of the deep linear neural networks [3] and the over-parameterized shallow neural networks [4] have the favorable characteristics that all local mins are global, and all saddles are strict. Based on the global landscape analysis, generic saddle-point escaping algorithms have been further developed, e.g., trust region method [5] and perturbed gradient descent [6]. However, these algorithms have either high iteration cost or iteration complexity, yielding conservative computational guarantees for specific problems [7]. Furthermore, these algorithms are more intricate than the vanilla gradient descent method.

Recently, the local landscape analysis turns out to be effective to enjoy fast convergence rate with cheap iteration cost via exploiting the local strong convexity and smoothness of the nonconvex loss functions [4, 8, 9, 10]. However, all these provable algorithms still call for carefully-designed initialization [4, 8, 9]. To find a natural implementation for the practitioners, in this paper, we shall propose to learn the shallow neural networks via *randomly initialized* gradient descent with provable optimality guarantees. Specifically, given enough training data, we show that the randomly initialized gradient descent iterates are able to enter a local region that enjoys strong convexity and strong smoothness within a few iterations. In the second stage, the gradient descent provably converges to a globally optimal model at a linear rate.

2. PROBLEM FORMULATION

In this paper, we consider a shallow neural network that consists of only one hidden layer with r neurons and activation function $\sigma(z) = z^2$ [4, 11, 12], n input nodes and one output node, as illustrated in Fig. 1. More precisely, the overall relationship among these layers is formulated by the following equation:

$$y = \sum_{i=1}^{r} \alpha_i \sigma(\langle \boldsymbol{w}_i, \boldsymbol{x} \rangle) = \sum_{j=1}^{r} \alpha_j \langle \boldsymbol{w}_j, \boldsymbol{x} \rangle^2, \qquad (1)$$

This work was supported in part by the National Nature Science Foundation of China under Grant 61601290, the Shanghai Sailing Program under Grant 16YF1407700 and the Hong Kong Research Grant Council under Grant 16211815.

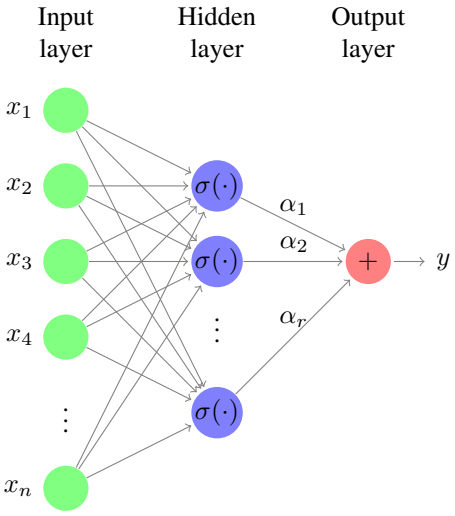


Fig. 1: Two-layer neural network with activation $\sigma(\cdot)$

where the scalar $y \in \mathbb{R}$ and the vector $\boldsymbol{x} \in \mathbb{R}^{n\times 1}$ represent the output and the input, respectively. $\boldsymbol{w}_i$ contains the weights of the edges connecting the input to the i^{th} hidden node and α_i is the weight of the edge connecting the i^{th} hidden node to the output. We focus on the "under-parameterized" regime where the number of hidden nodes is much less than the dimension of the inputs ($r \ll n$) [13].

Furthermore, we propose to jointly optimize α_j and $\boldsymbol{w}_j$ by defining $\boldsymbol{W} = \sum_{j=1}^{r} \alpha_j \boldsymbol{w}_j \boldsymbol{w}_j^\top$ [13]. By factorizing $\boldsymbol{W}$ as $\boldsymbol{W} = \boldsymbol{M}\boldsymbol{M}^\top$, model (1) can be rewritten as follows:

$$y = \sum_{j=1}^{r} \alpha_j \boldsymbol{x}^\top \boldsymbol{w}_j \boldsymbol{w}_j^\top \boldsymbol{x} = \boldsymbol{x}^\top \boldsymbol{M}\boldsymbol{M}^\top \boldsymbol{x} = \|\boldsymbol{x}^\top \boldsymbol{M}\|_2^2, \tag{2}$$

where $\boldsymbol{M} \in \mathbb{R}^{n\times r}$ ($r \ll n$) denotes the low-rank factor. The goal is to recover $\boldsymbol{W}$, or equivalently, the low-rank factor $\boldsymbol{M}$, from limited number of observations. This problem spans a variety of important practical applications from machine learning (this paper) to communication [14].

Taking $\{\boldsymbol{x}_i, y_i\}_{i=1}^m$ as training data, we need to solve the following nonconvex optimization problem to learn a neural network:

$$\min_{\boldsymbol{M}\in\mathbb{R}^{n\times r}} \mathcal{L}(\boldsymbol{M}) = \frac{1}{4m}\sum_{i=1}^{m}(y_i - \|\boldsymbol{x}_i^\top \boldsymbol{M}\|_2^2)^2. \tag{3}$$

Obviously, the problem is highly nonconvex due to the natural least-squares empirical risk formulation in the optimization variable $\boldsymbol{M}$.

Our goal is to demonstrate that gradient descent (GD) with *random initialization* can solve the highly nonconvex problem (3) with global optimality guarantees.

3. MAIN RESULT

3.1. Preliminaries

We denote by $\|\boldsymbol{m}\|_2$ the l_2-norm of a vector $\boldsymbol{m}$, and $\boldsymbol{M}^\top$ and $\|\boldsymbol{M}\|_F$ the transpose and the Frobenius norm of a matrix $\boldsymbol{M}$, respectively. The k^{th} largest singular value of a matrix $\boldsymbol{M}$ is denoted by $\sigma_k(\boldsymbol{M})$. The notation $f(n) \lesssim g(n)$ or $f(n) = O(g(n))$ (resp. $f(n) \gtrsim g(n)$) means that there exists a universal constant $c > 0$ such that $|f(n)| \le c|g(n)|$ (resp. $|f(n)| \ge c|g(n)|$).

In our analysis, we specify the metric used to assess the estimation error of the running iterates.

Definition 1. Since $(\boldsymbol{M}^\natural\boldsymbol{P})(\boldsymbol{M}^\natural\boldsymbol{P})^\top = \boldsymbol{M}^\natural\boldsymbol{M}^{\natural\top}$ for any orthonormal matrix $\boldsymbol{P} \in \mathbb{R}^{r\times r}$, $\boldsymbol{M}^\natural$ is recoverable up to orthonormal transforms.

$$\text{dist}(\boldsymbol{M}_t, \boldsymbol{M}^\natural) = \|\boldsymbol{M}_t\boldsymbol{Q}_t - \boldsymbol{M}^\natural\|_F, \tag{4}$$

where $\boldsymbol{Q}_t$ is given by

$$\boldsymbol{Q}_t := \text{argmin}_{\boldsymbol{P}\in\mathcal{O}^{r\times r}} \|\boldsymbol{M}_t\boldsymbol{P} - \boldsymbol{M}^\natural\|_F. \tag{5}$$

3.2. Algorithm and theoretical results

The algorithm studied herein is a combination of vanilla gradient descent and random initialization

$$\boldsymbol{M}_{t+1} = \boldsymbol{M}_t - \mu_t \nabla\mathcal{L}(\boldsymbol{M}_t), \tag{6}$$

where $\boldsymbol{M}_t$ denotes the estimate in the t^{th} iteration, μ_t is the step size/learning rate, and the gradient $\nabla\mathcal{L}(\boldsymbol{M})$ is given by

$$\nabla\mathcal{L}(\boldsymbol{M}) = \frac{1}{m}\sum_{i=1}^{m}(\|\boldsymbol{x}_i^\top \boldsymbol{M}\|_2^2 - y_i)\boldsymbol{x}_i\boldsymbol{x}_i^\top \boldsymbol{M}. \tag{7}$$

Moreover, we apply the random initialization, which means the columns of $\boldsymbol{M}_0$ composed of standard Gaussian entries, i.e. $\boldsymbol{m}_i^0 \overset{\text{i.i.d}}{\sim} \mathcal{N}(0, n^{-1}\boldsymbol{I}_n)$, for $i = 1, \cdots, r$.

Our main findings are summarized in the following theorem.

Theorem 1. *Given a data set of training pairs $\{\boldsymbol{x}_i, y_i\}_{i=1}^m$ with the inputs $\boldsymbol{x}_i \in \mathbb{R}^n \overset{\text{i.i.d}}{\sim} \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_n)$ and the labels $y_i \in \mathbb{R}$ generated from a planted two layer neural network model with r hidden neurons*

$$y_i = \|\boldsymbol{x}_i^\top \boldsymbol{M}^\natural\|_2^2,$$

where $\boldsymbol{M}^\natural \in \mathbb{R}^{n\times r}$ are the weights of the input-hidden layer. Suppose the sample complexity m and the step size μ_t obeys

$$m \gtrsim nr^4\kappa^3 \log^{13} m \quad \text{and} \quad \mu_t := \mu = \frac{c}{r^2\kappa^2\sigma_r^2(\boldsymbol{M}^\natural)},$$

where $\kappa = \frac{\sigma_1^2(\boldsymbol{M}^\natural)}{\sigma_r^2(\boldsymbol{M}^\natural)}$ *for some sufficiently small constant* c*. Then with high probability approaching one, there exits a sufficiently small constant* $0 < \delta < 1$ *and* $T_\delta = O(r^2 \log n)$ *such that the trajectory of gradient descent with random initialization can be divided into two stages:*

- ***Stage 1.*** *The iterates are capable of entering a local region with strong convexity and smoothness surrounding* $\boldsymbol{M}^\natural$ *within* $T_\delta = O(r^2 \log n)$ *iterations,*

$$\mathrm{dist}(\boldsymbol{M}_{T_\delta}, \boldsymbol{M}^\natural) \le \delta \frac{\sigma_r^2(\boldsymbol{M}^\natural)}{\|\boldsymbol{M}^\natural\|_F}.$$

- ***Stage 2.*** *The iterates converge linearly to* $\boldsymbol{M}^\natural$ *with a contraction rate* $1 - 0.5\mu\sigma_r^2(\boldsymbol{M}^\natural)$

$$\mathrm{dist}(\boldsymbol{M}_t, \boldsymbol{M}^\natural) \le (1 - 0.5\mu\sigma_r^2(\boldsymbol{M}^\natural))^{t-T_\delta} \cdot \delta \frac{\sigma_r^2(\boldsymbol{M}^\natural)}{\|\boldsymbol{M}^\natural\|_F}, \quad t > T_\delta$$

Here, the step size is taken to be a fixed constant throughout all iterations, and we reuse all data across all iterations (i.e. resampling is not required to establish this theorem). Even though Stage 1 may not enjoy linear convergence in terms of the relative error, it is of fairly short duration $O(r^2 \log n)$. After entering the local region, GD converges linearly to the globally optimal model $\boldsymbol{M}^\natural$. This implies that GD will take $O(r^2 \log(1/\epsilon))$ iterations to reach ϵ-accuracy. Taken collectively, the theorem shows that the iterations complexity of gradient descent with random initialization is $O\left(r^2 \log n + r^2 \log \frac{1}{\epsilon}\right)$. Moreover, our findings only require that the sample size satisfies $m \gtrsim nr^4\kappa^3 \mathrm{poly}\log(m)$ which is optimal up to some logarithmic factor.

Compared with other existing nonconvex approaches, we are able to guarantee near-optimal sampling complexity and computational complexity simultaneously. Specifically, [11] adopted a greedy learning strategy, and can only guarantee sublinear convergence rate. Iterative algorithms based on SVD methods proposed by [13] requires a fresh set of samples at every iteration, which is never executed in practice, and the sample complexity grows unbounded for exact recovery. Moreover, [8] provided the similar conclusions using gradient descent but with spectral initialization. In contrast, our initialization scheme is natural implementation for practitioners with random initialization.

Other works have also studied similar two-layer neural network with quadratic activations [4, 12]. However, they studied the optimization for an over-parameterized shallow neural network with quadratic activation, where r is larger than n.

4. ANALYSIS

In this section, we will provide intuitions regarding why our theorem is expected to work. We provide the outline of the proof here.We will present more details of the proof in the extended version.

Definition 2. To capture the signal-to-noise ratio of the running iterates, we define the signal component $\boldsymbol{M}_{t,\|} = [\boldsymbol{m}_{i,\|}^t]_{i=1}^r$ and perpendicular components $\boldsymbol{M}_{t,\perp} = [\boldsymbol{m}_{i,\perp}^t]_{i=1}^r$. For simplicity, we denote $\boldsymbol{m}_i^\natural$ (resp. $\boldsymbol{m}_i^t$) as the i^{th} column of the $\boldsymbol{M}^\natural$ (resp. $\boldsymbol{M}^t$).

$$\boldsymbol{m}_{i,\|}^t = \frac{\boldsymbol{e}_i^\top \boldsymbol{M}^{t\top} \boldsymbol{M}^\natural \boldsymbol{e}_i}{\|\boldsymbol{m}_i^\natural\|^2} \boldsymbol{m}^\natural, \tag{8}$$

$$\boldsymbol{m}_{i,\perp}^t = \boldsymbol{m}_i^t - \frac{\boldsymbol{e}_i^\top \boldsymbol{M}^{t\top} \boldsymbol{M}^\natural \boldsymbol{e}_i}{\|\boldsymbol{m}_i^\natural\|^2} \boldsymbol{m}_i^\natural, \tag{9}$$

where $\boldsymbol{e}_i$ is the i^{th} standard base.

Definition 3. In what follows, we focus our attention on the following two quantities that reflect the sizes of the preceding two components

$$\alpha_t := \sqrt{\frac{1}{r}\sum_{i=1}^r \|\boldsymbol{m}_{i,\|}^t\|^2}, \quad \beta_t := \sqrt{\frac{1}{r}\sum_{i=1}^r \|\boldsymbol{m}_{i,\perp}^t\|_2^2}. \tag{10}$$

4.1. Population dynamics

We first investigate the dynamics of the population gradient sequence (the case where we have infinite samples). Hence, the iterates $\{\boldsymbol{M}_t\}$ are constructed using the population gradient

$$\boldsymbol{M}_{t+1} = \boldsymbol{M}_t - \mu_t \nabla \mathcal{F}(\boldsymbol{M}_t).$$

Here, $\nabla \mathcal{F}(\boldsymbol{M})$ represents the population gradient given by

$$\nabla \mathcal{F}(\boldsymbol{M}) = [(\|\boldsymbol{M}_t\|_F^2 - \|\boldsymbol{M}^\natural\|_F^2)\boldsymbol{I}_n + 2(\boldsymbol{M}_t\boldsymbol{M}_t^\top - \boldsymbol{M}^\natural\boldsymbol{M}^{\natural\top})]\boldsymbol{M}_t,$$

which can be computed by $\nabla \mathcal{F}(\boldsymbol{M}) := \mathbb{E}\left[\nabla \mathcal{L}(\boldsymbol{M})\right]$ assuming that $\boldsymbol{M}$ and $\boldsymbol{x}_i$'s are independent. Without loss of generality, we assume $\boldsymbol{M}^\natural = [\boldsymbol{e}_1, \cdots, \boldsymbol{e}_1]$. Then we obtain the dynamics for both signal and perpendicular components

$$\boldsymbol{M}_{t+1,\|} = \boldsymbol{M}_{t,\|}\{\left[1 + \mu(3r - \|\boldsymbol{M}_t\|_F^2)\right]\boldsymbol{I}_r - 2\mu\boldsymbol{M}_t^\top\boldsymbol{M}_t\}, \tag{11a}$$

$$\boldsymbol{M}_{t+1,\perp} = \boldsymbol{M}_{t,\perp}\{\left[1 + \mu(r - \|\boldsymbol{M}_t\|_F^2)\right]\boldsymbol{I}_r - 2\mu\boldsymbol{M}_t^\top\boldsymbol{M}_t\}. \tag{11b}$$

For simplicity, we denote by

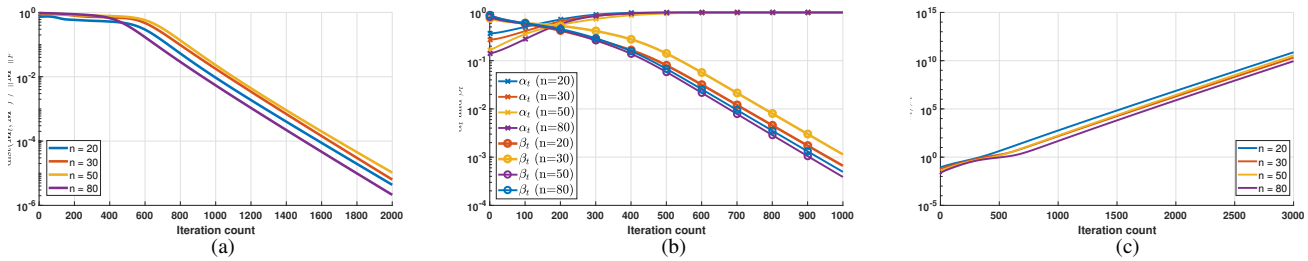
$$\boldsymbol{A} = \left[1 + \mu(3r - \|\boldsymbol{M}_t\|_F^2)\right]\boldsymbol{I}_r - 2\mu\boldsymbol{M}_t^\top\boldsymbol{M}_t, \tag{12a}$$

$$\boldsymbol{B} = \left[1 + \mu(r - \|\boldsymbol{M}_t\|_F^2)\right]\boldsymbol{I}_r - 2\mu\boldsymbol{M}_t^\top\boldsymbol{M}_t. \tag{12b}$$

Assuming $\boldsymbol{M}_t$ is non-negative definite, we arrive at the following population-level state evolution for both α_t and β_t:

$$\sqrt{r}\sigma_r(\boldsymbol{A})\alpha_t \le \alpha_{t+1} \le \sqrt{r}\sigma_1(\boldsymbol{A})\alpha_t, \tag{13a}$$

$$\sqrt{r}\sigma_r(\boldsymbol{B})\beta_t \le \beta_{t+1} \le \sqrt{r}\sigma_1(\boldsymbol{B})\beta_t. \tag{13b}$$



(a) (b) (c)

Fig. 2: Numerical results, plotted semilogarithmically

4.2. Finite-sample analysis

In the finite-sample regime, we rewrite the gradient update rule as

$$\begin{aligned}\boldsymbol{M}_{t+1} &= \boldsymbol{M}_t - \mu\nabla\mathcal{L}(\boldsymbol{M}_t) \\ &= \boldsymbol{M}_t - \mu\nabla\mathcal{F}(\boldsymbol{M}_t) - \mu r(\boldsymbol{M}_t),\end{aligned} \tag{14}$$

where $\boldsymbol{r}(\boldsymbol{M}_t) = \nabla\mathcal{L}(\boldsymbol{M}_t) - \nabla\mathcal{F}(\boldsymbol{M}_t)$. By assuming the independence between $\boldsymbol{M}_t$ and $\{\boldsymbol{x}_i\}$, the central limit theorem (CLT) allows us to control the size of the fluctuation term $\boldsymbol{r}(\boldsymbol{M}_t)$ as long as the sample size $m \gtrsim nr^4\text{poly}\log(m)$. Then we arrive at an approximation state evolution for the finite-sample case:

$$\sqrt{r}\sigma_r(\boldsymbol{A})\alpha_t \lesssim \alpha_{t+1} \lesssim \sqrt{r}\sigma_1(\boldsymbol{A})\alpha_t, \tag{15a}$$

$$\sqrt{r}\sigma_r(\boldsymbol{B})\beta_t \lesssim \beta_{t+1} \lesssim \sqrt{r}\sigma_1(\boldsymbol{B})\beta_t. \tag{15b}$$

4.3. Outline of the proof

When $|\alpha_t - 1| \le \delta/2r$ and $|\beta_t| \le \delta/2r$, then it is easy to check that

$$\text{dist}(\boldsymbol{M}_t, \boldsymbol{M}^\natural) \le \sqrt{r}|\alpha_t - 1| + \sqrt{r}|\beta_t| \le \delta/\sqrt{r}.$$

1. Show that if α_t and β_t satisfy the approximate state evolution (15), then there exists some $T_\delta = O(r^2\log(n))$ such that

$$|\alpha_{T_\delta} - 1| \le \delta/2r \;\text{ and }\; |\beta_{T_\delta}| \le \delta/2r, \tag{16}$$

   which immediately implies that

$$\text{dist}(\boldsymbol{M}_t, \boldsymbol{M}^\natural) \le \delta/\sqrt{r}.$$

2. Justify that α_t and β_t satisfy the approximate state evolution with high probability, using leave-one-out arguments [7].

After $t \ge T_\delta$, we can invoke prior theory [8] concerning local convergence to show that with high probability,

$$\text{dist}(\boldsymbol{M}_t, \boldsymbol{M}^\natural) \le (1-\rho)^{t-T_\delta}\|\boldsymbol{M}_t - \boldsymbol{M}^\natural\|_F, \;\; \forall t \ge T_\delta,$$

for some constant $0 < \rho < 1$.

5. NUMERICAL RESULTS

We now provide numerical results that shed some more light to the conclusions drawn from Theorem 1.

Let us place ourselves under the setting of Theorem 1. We vary the number n of dimensions (i.e. n = 20, 30, 50, 80), set $m = 1000n$, fix $r = 10$ and take a constant step size $\mu := 0.005$. Here the design vectors are generated from Gaussian distributions, i.e., $\boldsymbol{x}_i \overset{i.i.d}{\sim} \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_n)$ for $1 \le i \le m$. Without loss of generality, we normalize the columns of $\boldsymbol{M}^\natural$ with the length of 1. We use metric (4) to evaluate the performance. Fig. 2(a) displays the convergence results of gradient descent with random initialization and a constant step size: Stage 1, the relative error of $\boldsymbol{M}_t$ stays nearly flat; Stage 2, the relative error of $\boldsymbol{M}_t$ experiences geometric decay. Importantly, the first stage lasts only a few hundred of iterations.

In Fig. 2(b), the size of the signal component increases exponentially fast and becomes the dominant component within several hundreds of iterations. Furthermore, we find the ratio α_t/β_t grows exponentially fast throughout the execution of the algorithm, as illustrated in Fig .2(c). The ratio α_t/β_t in some sense captures the signal-to-noise ratio of the running iterates.

6. CONCLUSION

We demystified the computational efficiency of gradient descent with random initialization for learning a shallow neural network with quadratic activations. Specifically, we demonstrated that gradient descent with random initialization takes only $O\left(r^2\log n + r^2\log\frac{1}{\epsilon}\right)$ iterations to converge a globally optimal model given nearly minimal samples.

7. REFERENCES

[1] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio, *Deep learning*, vol. 1, MIT press Cambridge, 2016.

[2] Yuchen Zhang, Percy Liang, and Martin J Wainwright, "Convexified convolutional neural networks," *Int. Conf. Mach. Learn. (ICML)*, pp. 4044–4053, 2017.

[3] Kenji Kawaguchi, "Deep learning without poor local minima," *Adv. Neural. Inf. Process. Syst. (NIPS)*, pp. 586–594, 2016.

[4] Mahdi Soltanolkotabi, Adel Javanmard, and Jason D Lee, "Theoretical insights into the optimization landscape of over-parameterized shallow neural networks," *IEEE Trans. Inf. Theory*, 2018.

[5] Yann N Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio, "Identifying and attacking the saddle point problem in high-dimensional non-convex optimization," *Adv. Neural. Inf. Process. Syst. (NIPS)*, pp. 2933–2941, 2014.

[6] Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan, "How to escape saddle points efficiently," *Int. Conf. Mach. Learn. (ICML)*, pp. 724–1732, 2017.

[7] Yuxin Chen, Yuejie Chi, Jianqing Fan, and Cong Ma, "Gradient descent with random initialization: Fast global convergence for nonconvex phase retrieval," *arXiv preprint arXiv:1803.07726*, 2018.

[8] Yuanxin Li, Cong Ma, Yuxin Chen, and Yuejie Chi, "Nonconvex matrix factorization from rank-one measurements," *arXiv preprint arXiv:1802.06286*, 2018.

[9] Sanjeev Arora, Nadav Cohen, Noah Golowich, and Wei Hu, "A convergence analysis of gradient descent for deep linear neural networks," *arXiv preprint arXiv:1810.02281*, 2018.

[10] J. Dong and Y. Shi, "Nonconvex demixing from bilinear measurements," *IEEE Trans. Signal Process.*, vol. 66, no. 19, pp. 5152–5166, Oct. 2018.

[11] Roi Livni, Shai Shalev-Shwartz, and Ohad Shamir, "On the computational efficiency of training neural networks," *Adv. Neural. Inf. Process. Syst. (NIPS)*, pp. 855–863, 2014.

[12] Yuanzhi Li, Tengyu Ma, and Hongyang Zhang, "Algorithmic regularization in over-parameterized matrix sensing and neural networks with quadratic activations," *Conf. On Learning Theory (COLT)*, pp. 2–47, 2018.

[13] Mohammadreza Soltani and Chinmay Hegde, "Towards provable learning of polynomial neural networks using low-rank matrix estimation," *Int. Conf. Art. Intell. Stat. (AISTATS)*, pp. 1417–1426, 2018.

[14] Jialin Dong, Kai Yang, and Yuanming Shi, "Blind demixing for low-latency communication," *IEEE Transactions on Wireless Communications*, 2018.