Towards Understanding the Importance of Noise in Training Neural Networks

Mo Zhou [*1] Tianyi Liu [*2] Yan Li [2] Dachao Lin [1] Enlu Zhou [2] Tuo Zhao [2]

Abstract

Numerous empirical evidence has corroborated that noise plays a crucial rule in effective and efficient training of neural networks. The theory behind, however, is still largely unknown. This paper studies this fundamental problem through training a simple two-layer convolutional neural network model. Although training such a network requires solving a non-convex optimization problem with a spurious local optimum and a global optimum, we prove that perturbed gradient descent and perturbed mini-batch stochastic gradient algorithms in conjunction with noise annealing is guaranteed to converge to a global optimum in polynomial time with arbitrary initialization. This implies that the noise enables the algorithm to efficiently escape from the spurious local optimum. Numerical experiments are provided to support our theory.

1. Introduction

Deep neural networks (DNNs) have achieved great successes in a wide variety of domains such as speech and image recognition (Hinton et al., 2012; Krizhevsky et al., 2012), nature language processing (Rumelhart et al., 1986) and recommendation systems (Salakhutdinov et al., 2007). Training DNNs requires solving non-convex optimization problems. Specifically, given n samples denoted by $\{(x_i, y_i)\}_{i=1}^n$, where x_i is the i-th input feature and y_i is the response, we solve the following optimization problem,

$$\min_\theta \mathcal{F}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(x_i, \theta)),$$

where ℓ is a loss function, f denotes the decision function based on the neural network, and θ denotes the parameters associated with f.

Simple first order algorithms such as Stochastic Gradient Descent (SGD) and its variants have been very successful for training deep neural networks, despite the highly complex non-convex landscape. For instance, recent results show that there are a combinatorially large number of saddle points and local optima in training DNNs (Choromanska et al., 2015). Though it has been shown that SGD and its variants can escape saddle points efficiently and converge to local optima (Dauphin et al., 2014; Kawaguchi, 2016; Hardt & Ma, 2016; Jin et al., 2017), the reason why the neural network learnt by SGD generalizes well cannot yet be well explained, since local optima do not necessarily guarantee generalization. For example, Zhang et al. (2016) empirically show the proliferation of global optima (when minimizing the empirical risk), most of which cannot generalize; Keskar et al. (2016) also provide empirical evidence of the existence of sharp local optima, which do not generalize. They further observe that gradient descent (GD) can often converge to the sharp optima, while SGD tends to converge to the flat ones. This phenomenon implies that the noise in SGD is very crucial and enables SGD to select good optima. Besides, Bottou (1991); Neelakantan et al. (2015) also show that adding noise to gradient can potentially improve training of deep neural networks. These empirical observations motivate us to theoretically investigate the role of the noise in training DNNs.

This paper aims to provide more theoretical insights on the following fundamental question:

How does noise help train neural networks in the presence of bad local optima?

Specifically, we study a two-layer non-overlapping convolutional neural network (CNN) with one hidden layer, which takes the following form:

$$f(Z, w, a) = a^T \sigma(Z^T w),$$

where $w \in \mathbb{R}^p$, $a \in \mathbb{R}^k$ and $Z \in \mathbb{R}^{p\times k}$ are the convolutional weights, the output weights and the input, respectively, and $\sigma(\cdot)$ is the element-wise ReLU activation operator. Since the ReLU activation is positive homogeneous, the weights a and w can arbitrarily scale with each other. Thus, we impose an additional constraint $\|w\|_2 = 1$ to make the neural network identifiable. We consider the realizable case, where the training data is generated from a **teacher** network

[*]Equal contribution [1]Peking University [2]Georgia Institute of Technology. Correspondence to: Tianyi Liu and Tuo Zhao <{tianyiliu,tourzhao}@gatech.edu>.

Proceedings of the 36th International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

with true parameters w^*, a^* and $\|w^*\|_2 = 1$. Then we aim to recover the **teacher** neural network by solving the following optimization problem,

$$\min_{w,\, a} \ \frac{1}{2}\mathbb{E}_Z\left(f(Z,w,a) - f(Z,w^*,a^*)\right)^2 \qquad \text{subject to} \quad w^\top w = 1, \tag{1}$$

where Z is independent Gaussian input[1]. One can verify that (w^*, a^*) is a global optimum of (1).

Though over-simplified compared with complex deep neural networks in practice, the above model turns out to have some intriguing properties, which helps us get insight into understanding the optimization landscape of training neural networks. Specifically, Du et al. (2017) show that the optimization problem (1) has a non-trivial spurious optimum, which does not generalize well. They further prove that with random initialization, Gradient Descent (GD) can be trapped in this spurious optimum with constant probability[2].

Inspired by Du et al. (2017), we propose to investigate whether adding noise to gradient descent helps avoid the spurious optimum using the same model. Specifically, we consider a perturbed GD algorithm[3] in conjunction with noise annealing to solve the optimization problem (1). To be more concrete, we run the algorithm with multiple epochs and decrease the magnitude of the noise as the number of epochs increases. Note that our algorithm is different from SGD in terms of the noise. In our algorithm, we inject independent noise to the gradient update at every iteration, while the noise of SGD comes from the training sample. As a consequence, the noise of SGD has very complex dependence on the iterate, which is very difficult to analyze. See more detailed discussions in Sections 2 and 6.

We further analyze the convergence properties of our perturbed GD algorithm: At early stages, large noise essentially convolutes with the loss surface and makes the optimization landscape smoother, which tames non-convexity and rules out the spurious local optimum. Hence, perturbed GD is capable of escaping from the spurious local optimum. Though large noise leads to large optimization errors, this can be further compensated by noise annealing. In another word, the injected noise with decreasing magnitude essentially guides GD to gradually approach and eventually fall in the basin of attraction of the global optimum. Given that the noise has been annealed to a sufficiently small level at later stages, the algorithm finally converges to the global optimum and stays in its neighborhood. Overall, we prove that with random initialization and noise annealing, perturbed GD is guaranteed to converge to the global optimum with high probability in polynomial time. Moreover, we further extend our proposed theory to the perturbed mini-batch stochastic gradient algorithm, and establish similar theoretical guarantees. To the best of our knowledge, this is the first theoretical result towards justifying the effect of noise in training NNs by first order algorithms in the presence of the spurious local optima.

Our work is related to Zhou et al. (2017); Li & Yuan (2017); Kleinberg et al. (2018); Jin et al. (2018), which also study the effect of noise in non-convex optimization. We give detailed discussions in Section 6.

The rest of the paper is organized as follows: Section 2 describes the two-layer non-overlapping convolutional network and introduces our perturbed GD algorithm; Section 3, 4 present the convergence analysis; Section 5 provides the numerical experiments; Section 6 discusses related works.

Notations: Given a vector $v = (v_1, \ldots, v_d)^\top \in \mathbb{R}^d$, we define $\|v\|_2^2 = \sum_j v_j^2$, $\|v\|_1 = \sum_j |v_j|$. For vectors $v, u \in \mathbb{R}^d$, we define $\langle u, v\rangle = \sum_{j=1}^d u_j v_j$. $\mathbb{B}_0(r)$ denotes a ball with radius r centered at zero in $\mathbb{R}^d$, i.e., $\mathbb{B}_0(r) = \{v \in \mathbb{R}^d \mid \|v\|_2 \le r\}$ and $\mathbb{S}_0(r)$ denotes the boundary of $\mathbb{B}_0(r)$. For two vectors v, w, $\angle(v,w)$ represents the angle between them, i.e., $\angle(v,w) = \arccos\frac{v^\top w}{\|v\|_2\|w\|_2}$. We denote the uniform distribution on $\mathcal{M} \subseteq \mathbb{R}^d$ by $\mathrm{unif}(\mathcal{M})$ and the projection of vector v on set $\mathcal{M}$ by $\mathrm{Proj}_{\mathcal{M}}(v)$. For two sets A and $B \in \mathbb{R}^d$, $A \backslash B = \{x \in \mathbb{R}^d \big| x \in A,\ x \notin B\}$.

2. Model and Algorithm

We first introduce the neural network models of our interests, and then present the nonconvex optimization algorithm.

2.1. Neural Network Models

Recall that we study a two-layer non-overlapping convolutional neural network (CNN) given by:

$$f(Z, w, a) = a^\top \sigma(Z^\top w), \tag{2}$$

where $a \in \mathbb{R}^k$, $w \in \mathbb{R}^p$ and $Z \in \mathbb{R}^{p\times k}$ are the output weights, the convolutional weights and input, respectively. $\sigma(\cdot)$ denotes the element-wise ReLU activation operator. Since the ReLU activation is homogeneous, w and a can arbitrarily scale with each other without changing the output of the network, i.e., $f(Z, w, a) = f(Z, cw, \frac{a}{c})$ for any $c > 0$. Thus, we impose an additional constraint $\|w\|_2 = 1$ to make the model identifiable. We assume independent Gaussian input $Z = [Z_1, ..., Z_p]$, where Z_i's are independently sampled from $N(0, I)$, and focus on the noiseless realizable setting – i.e., the response is generated by a noiseless **teacher** network

$$y = f(Z, w^*, a^*) = (a^*)^\top \sigma(Z^\top w^*)$$

[1] This is a common assumption in previous works (Tian, 2017; Brutzkus & Globerson, 2017; Zhong et al., 2017)

[2] Du et al. (2017) prove that this probability is bounded between $1/4$ and $3/4$. Their numerical experiments show that this probability can be as worse as $1/2$.

[3] Our algorithm actually updates w using the manifold gradient over the sphere. See more details in Section 2

with some true parameters $\|w^*\|_2 = 1$ and a^*. We aim to learn a **student** network, i.e., recover the true parameters (w^*, a^*) by solving the following regression problem using mean square loss:

$$\min_{w,a} \mathcal{L}(w, a) \quad \text{subject to} \quad w^\top w = 1, \tag{3}$$

where $\mathcal{L}(w, a) = \frac{1}{2}\mathbb{E}_Z(f(Z, w, a) - f(Z, w^*, a^*))^2$. The optimization landscape has been partially studied by Du et al. (2017). Specifically, one can easily verify that (w^*, a^*) is a global optimum of (3). Moreover, they prove that there exists a spurious local optimum, and gradient descent with random initialization can be trapped in this spurious optimum with constant probability.

Proposition 1 (Informal, Du et al. (2017))**.** *Given*

$$w_0 \sim \text{unif}\big(\mathbb{S}_0\,(1)\,\big) \quad \textit{and} \quad a_0 \sim \text{unif}\big(\mathbb{B}_0\big(|\mathbf{1}^\top a^*|/\sqrt{k}\big)\big)$$

as the initialization and the learning rate is sufficiently small, then with at least probability $1/4$, *GD converges to the spurious local minimum* $(v^*, \widetilde{a})$ *satisfying*

$$\angle(v^*, w^*) = \pi, \ \widetilde{a} = (\mathbf{1}\mathbf{1}^\top + (\pi - 1)I)^{-1}(\mathbf{1}\mathbf{1}^\top - I)a^*.$$

Please refer to Du et al. (2017) for more details.

2.2. Optimization Algorithm

We then present the perturbed gradient descent algorithm for solving (3). Specifically, at the $t-$th iteration, we perturb the iterate (w_t, a_t) with independent noise $\xi_t \sim \text{unif}\big(\mathbb{B}_0(\rho_w)\big)$ and $\epsilon_t \sim \text{unif}\big(\mathbb{B}_0(\rho_a)\big)$ and take:

$$\begin{aligned} a_{t+1} &= a_t - \eta\nabla_a\mathcal{L}(w_t + \xi_t, a_t + \epsilon_t), \\ w_{t+1} &= \text{Proj}_{\mathbb{S}_0(1)}\big(w_t - \eta(I - w_t w_t^\top) \\ &\qquad\qquad \cdot \nabla_w\mathcal{L}(w_t + \xi_t, a_t + \epsilon_t)\big), \end{aligned}$$

where η is the learning rate. We remark that the update for w in our algorithm is essentially based on the manifold gradient, where $(I - w_t w_t^\top)$ is the projection operator to the tangent space of the unit sphere at w_t. For simplicity, we still refer to our algorithms as Perturbed Gradient Descent.

As can be seen, for $\xi_t = 0$ and $\epsilon_t = 0$, our algorithm is reduced to the (noiseless) gradient descent. Different from SGD, the noise of which is usually from randomly sampling the data, we inject the noise directly to the iterate used for computing gradient. Moreover, stochastic gradient is usually an unbiased estimate of gradient, while our perturbed gradient $\nabla_a L(w_t + \xi_t, a_t + \epsilon_t)$ and $\nabla_w L(w_t + \xi_t, a_t + \epsilon_t)$ yield biased estimates, i.e.,

$$\begin{aligned} \mathbb{E}_{\xi_t,\epsilon_t}\nabla_a\mathcal{L}(w_t + \xi_t, a_t + \epsilon_t) &\neq \nabla_a\mathcal{L}(w_t, a_t), \\ \mathbb{E}_{\xi_t,\epsilon_t}\nabla_w\mathcal{L}(w_t + \xi_t, a_t + \epsilon_t) &\neq \nabla_w\mathcal{L}(w_t, a_t). \end{aligned}$$

See detailed discussions in Section 6 and Appendix A.

Our algorithm also incorporates the noise annealing approach. Specifically, the noise annealing consists of multiple epochs with varying noise levels. Specifically, we use large noise in early epochs and gradually decrease the noise level as the number of epoch increases. Since we sample the noise ξ_t and ϵ_t uniformly from $\mathbb{B}_0(\rho_w)$ and $\mathbb{B}_0(\rho_a)$, respectively, we can directly control the noise level by controlling the radius of the ball, i.e., ρ_w and ρ_a. One can easily verify

$$\begin{aligned} \|\xi_t\|_2 \leq \rho_w, \ \mathbb{E}\xi_t = 0, \ \text{Cov}\,\xi_t = \frac{\rho_w^2}{p+2}I, \\ \|\epsilon_t\|_2 \leq \rho_a, \ \mathbb{E}\epsilon_t = 0, \ \text{Cov}\,\epsilon_t = \frac{\rho_a^2}{k+2}I. \end{aligned}$$

We summarize the algorithm in Algorithm 1.

Remark 2. *Note that our arbitrary initialization is different from the random initialization in Du et al. (2017), which requires* $w_0 \sim \text{unif}\,(\mathbb{S}_0(1))$ *and* $a_0 \sim \text{unif}\left(\mathbb{B}_0\left(\frac{|\mathbf{1}^\top a^*|}{\sqrt{k}}\right)\right)$. *They need the randomness to avoid falling into the basin of attraction of the spurious local optimum. Our perturbed GD, however, can be guaranteed to escape the spurious local optimum. Thus, we initialize the algorithm arbitrarily.*

Remark 3 (Convolutional Effects)**.** *We remark that the* s*-epoch of the perturbed GD can also be viewed as solving*

$$\min_{\|w\|_2=1,a} \mathbb{E}_{\xi_s,\epsilon_s}\mathcal{L}(w + \xi_s, a + \epsilon_s), \tag{4}$$

where $\xi_s \sim \text{unif}\big(\mathbb{B}_0(\rho_w^s)\big)$ *and* $\epsilon_s \sim \text{unif}\big(\mathbb{B}_0(\rho_a^s)\big)$. *Therefore, the noise injection can be interpreted as convoluting the objective function with uniform kernels. Such a convolution makes the objective much smoother, and leads to a benign optimization landscape with respect to the global optimum of the original problem, as illustrated in Figure 1 (See more details in the next section).*

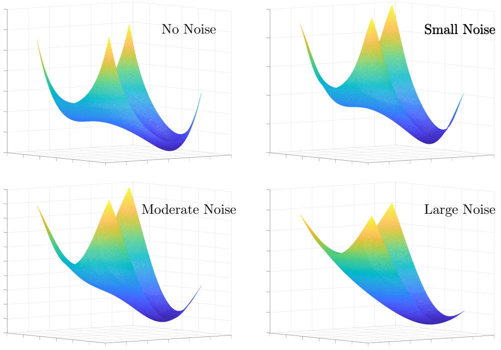


Figure 1. An illustration of the convolutional effects of the injected noise. Larger noise leads to smoother optimization landscapes, but also yields larger approximation errors to the original problem.

Note that the above convolution effect also introduces additional "bias" and "variance": (I) The global optimum of the smooth approximation (4) is different from the original problem; (II) The injected noise prevents the algorithm from converging. This is why we need to gradually decreasing the magnitude of the noise, which essentially guides the perturbed GD to gradually approach and eventually fall in the basin of attraction of the global optimum of the original problem (as illustrated in Figure 2).

Algorithm 1 *Perturbed Gradient Descent Algorithm with Noise Annealing*

```
input: number of epochs $S$, length of epochs $\{T_s\}_{s=1}^S$, learning rate schedule $\{\eta_s\}_{s=1}^S$ and noise level schedule $\{\rho_w^s\}_{s=1}^S$, $\{\rho_a^s\}_{s=1}^S$
initialize: choose any $w_0 \in \mathbb{S}_0(1)$ and $a_0 \in \mathbb{B}_0\big(\frac{|\mathbf{1}^\top a^*|}{\sqrt{k}}\big)$
for $s = 1, \dots, S$ do
    $w_{s,1} \leftarrow w_0$, $a_{s,1} \leftarrow a_0$
    for $t = 1 \dots T_s - 1$ do
        $\xi_{s,t} \sim \mathrm{unif}\big(\mathbb{B}_0(\rho_w^s)\big)$ and $\epsilon_{s,t} \sim \mathrm{unif}\big(\mathbb{B}_0(\rho_a^s)\big)$
        $a_{s,t+1} \leftarrow a_{s,t} - \eta_s \nabla_a \mathcal{L}(w_{s,t} + \xi_{s,t}, a_{s,t} + \epsilon_{s,t})$
        $w_{s,t+1} \leftarrow \mathrm{Proj}_{\mathbb{S}_0(1)}\big(w_{s,t} - \eta_s (I - w_{s,t} w_{s,t}^\top)$
            $\cdot \nabla_w \mathcal{L}(w_{s,t} + \xi_{s,t}, a_{s,t} + \epsilon_{s,t})\big)$
    $w_0 \leftarrow w_{s,T_s}$, $a_0 \leftarrow a_{s,T_s}$
output: $(w_{s,T_s},\ a_{s,T_s})$
```

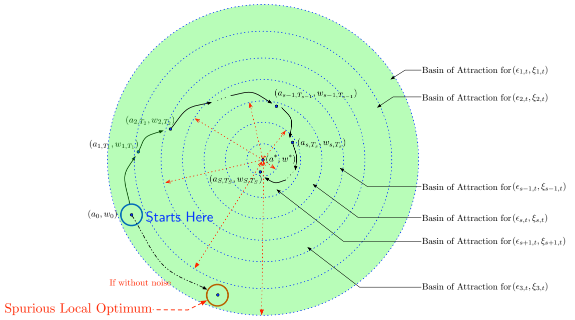

Basin of Attraction for $(\epsilon_{1,t}, \xi_{1,t})$
Basin of Attraction for $(\epsilon_{2,t}, \xi_{2,t})$
$(a_{s-1,T_{s-1}}, w_{s-1,T_{s-1}})$
(a_{2,T_2}, w_{2,T_2})
(a_{1,T_1}, w_{1,T_1})
(a_{s,T_s}, w_{s,T_s})
(a^*, w^*)
(a_{S,T_S}, w_{S,T_S})
Basin of Attraction for $(\epsilon_{s-1,t}, \xi_{s-1,t})$
(a_0, w_0)
Starts Here
Basin of Attraction for $(\epsilon_{s,t}, \xi_{s,t})$
Basin of Attraction for $(\epsilon_{s+1,t}, \xi_{s+1,t})$
If without noise
Basin of Attraction for $(\epsilon_{3,t}, \xi_{3,t})$
Spurious Local Optimum

Figure 2. An illustration of the noise injection in the perturbed GD algorithm. The injected noise with decreasing magnitude essentially guides the perturbed GD to gradually approach and eventually fall in the basin of attraction of the global optimum.

3. Convergence Analysis

We investigate the algorithmic behavior of the proposed perturbed GD algorithm. Our analysis shows that the noise injected to the algorithm has a convolutional effect on the loss surface and makes the optimization landscape smoother, which tames non-convexity by avoiding being trapped at the bad local optimum. Thus, our proposed algorithm can converge to the global one.

Our theory essentially reveals a phase transition as the magnitude of the injected noise decreases. For simplicity, our analysis only considers a two-epoch version of the proposed perturbed GD algorithm, but can be generalized to the multiple-epoch setting (See more detailed discussions in Section 6). Specifically, the first epoch corresponds to **Phase I**, and the proposed algorithm shows an escaping from the spurious local optimum phenomenon, as the injected noise is sufficiently large; The second epoch corresponds to **Phase II**, and the proposed algorithm demonstrates convergence to the global optimum, as the injected noise is reduced.

Before we proceed with our main results, we first define the partial dissipative condition for an operator $\mathcal{H}$ as follows.

Definition 4 (Partial dissipativity)**.** *Let $\mathcal{M}$ be a subset of $\{1, 2, ..., d\}$ with $|\mathcal{M}| = m$, and $x_{\mathcal{M}}$ be the subvector of $x \in \mathbb{R}^d$ with all indices in $\mathcal{M}$. For any operator $\mathcal{H} : \mathbb{R}^d \to \mathbb{R}^m$, we say that $\mathcal{H}$ is $(c_{\mathcal{M}}, \gamma_{\mathcal{M}}, \mathcal{M})$-partial dissipative with respect to (w.r.t.) the subset $\mathcal{X}^* \subseteq \mathbb{R}^d$ over the set $\mathcal{X} \supseteq \mathcal{X}^*$, if for every $x \in \mathcal{X}$, there exist an $x^* \in \mathcal{X}^*$ and two positive universal constants $c_{\mathcal{M}}$ and $\gamma_{\mathcal{M}}$ such that*

$$\langle -\mathcal{H}(x), x^*_{\mathcal{M}} - x_{\mathcal{M}} \rangle \geq c_{\mathcal{M}} \|x_{\mathcal{M}} - x^*_{\mathcal{M}}\|_2^2 - \gamma_{\mathcal{M}}. \quad (5)$$

$\mathcal{X}$ is called the partial dissipative region of the operator $\mathcal{H}$ w.r.t. $x_{\mathcal{M}}$.

The partial dissipativity in definition 4 is actually a generalization of the joint dissipativity from existing literature on studying attractors of dynamical systems (Barrera & Jara, 2015). To be specific, when $\mathcal{M} = \{1, 2, ..., d\}$, partial dissipativity is reduced to dissipativity. Here we are using the partial dissipativity, since our proposed algorithm can be viewed as a complicated dynamical system, and the global optimum is the target attractor.

The variational coherence studied in (Zhou et al., 2017) and one point convexity studied in (Kleinberg et al., 2018) can be viewed as the special example of partial dissipativity. Specifically, they consider $\gamma = 0$, the operator $\mathcal{H}$ as the gradient of the objective function f and $\mathcal{X}^*$ as the set of all minimizers of f. More precisely, their conditions require

$$\langle -\nabla f(x), x^* - x \rangle > c\|x - x^*\|_2^2,$$

i.e., the negative gradient of the objective function to have a positive fraction pointing toward $\mathcal{X}^*$, and therefore the gradient descent algorithm is guaranteed to make progress towards the optimum at every iteration. The variational coherence/one point convexity, though nice and intuitive, is a very strong assumption. For the optimization problem of our interest in (3), such a condition does not hold even within a small neighborhood around the global optimum. Fortunately, we show that the problem enjoys partial dissipativity which is more general and can characterize more complicated structure of the problem. Please see more discussion in Section 6.

3.1. Phase I: Escaping from the Local Optimum

We first characterize the algorithmic behavior of our proposed algorithm in Phase I. Note that our proposed perturbed GD algorithm, different from GD, intentionally injects noise at each iteration, and the update is essentially based on the perturbed gradient. The following theorem characterizes the partial dissipativity of the perturbed gradient.

Theorem 5. *Choose $\rho_w^0 = C_w^0 kp^2 \geq 1$ and $\rho_a^0 = C_a^0$ for large enough constants C_w and C_a. Let $\xi \sim \mathrm{unif}(\mathbb{B}_0(\rho_w^0))$ and $\epsilon \sim \mathrm{unif}(\mathbb{B}_0(\rho_a^0))$. There exist some constants C_1 and C_2 such that the perturbed gradient of L w.r.t. a satisfies*

$$\langle -\mathbb{E}_{\xi,\epsilon}\nabla_a \mathcal{L}(w+\xi, a+\epsilon), a^* - a\rangle \geq \frac{C_1}{p}\|a - a^*\|_2^2$$

for any $(w, a) \in \mathcal{A}_{C_2,C_3}$, where

$$\begin{aligned}\mathcal{A}_{C_2,C_3} = \big\{(w,a)\ \big|a^\top a^* \leq \frac{C_2}{p}\|a^*\|_2^2 \text{ or}\\ \|a - a^*/2\|_2^2 \geq \|a^*\|_2^2,\ w \in \mathbb{S}_0(1),\\ -4(\mathbf{1}^\top a^*)^2 \leq \mathbf{1}^\top a^* \mathbf{1}^\top a - (\mathbf{1}^\top a^*)^2 \leq \frac{C_3}{p}\|a^*\|_2^2\big\}.\end{aligned}$$

Moreover, for any $C_4 \in (-1, 1]$ and $M > m > 0$, there exists some constant C_5 such that the perturbed manifold gradient of L w.r.t. w satisfies

$$\begin{aligned}&\langle -\mathbb{E}_{\xi,\epsilon}(I - ww^\top)\nabla_w \mathcal{L}(w+\xi, a+\epsilon), w^* - w\rangle\\ &\geq \frac{m(1+C_4)}{16}\|w - w^*\|_2^2 - C_5\frac{k}{\rho_w},\end{aligned}$$

for any $(w, a) \in \mathcal{K}_{C_4,m,M}$, where

$$\begin{aligned}\mathcal{K}_{C_4,m,M} = \Big\{(w,a)\ \big|\ a^\top a^* \in [m, M],\\ w^\top w^* \geq C_4,\ w \in \mathbb{S}_0(1)\Big\}.\end{aligned}$$

The detailed proof of Theorem 5 is provided in Appendix C.1. Theorem 5 shows that the partial dissipativity holds for the perturbed gradient of L with respect to a over $\mathcal{A}_{C_2,C_3}$, and the partial dissipativity holds for the perturbed manifold gradient of L with respect to w over $\mathcal{K}_{C_4,m,M}$, respectively. Note that the joint dissipativity can hold but only over a smaller set $\mathcal{A}_{C_2,C_3} \cap \mathcal{K}_{C_4,m,M}$. Fortunately, the partial dissipativity is enough to ensure our proposed algorithm to make progress at every iteration, even though the joint dissipativity does not hold. As a result, our proposed algorithm can avoid being trapped by the spurious local optimum. For simplicity, we denote ϕ_t as the angle between w_t and w^*, i.e., $\phi_t = \angle(w_t, w^*)$. The next theorem analyzes the algorithmic behavior of perturbed GD in Phase I.

Theorem 6. *Suppose $\rho_w^0 = C_w^0 kp^2 \geq 1$, $\rho_a^0 = C_a^0$, $a_0 \in \mathbb{B}_0\Big(\frac{|\mathbf{1}^\top a^*|}{\sqrt{k}}\Big)$ and $w_0 \in \mathbb{S}_0(1)$. For any $\delta \in (0, 1)$, we choose step size*

$$\eta = C_6\Big(k^4 p^6 \cdot \max\Big\{1, p\log\frac{1}{\delta}\Big\}\Big)^{-1}$$

for some constant C_6. Then with at least probability $1-\delta$, we have

$$m_a \leq a_t^\top a^* \leq M_a \quad \textit{and} \quad \phi_t \leq \frac{5}{12}\pi \tag{6}$$

for all $T_1 \leq t \leq \widetilde{O}(\eta^{-2})$, where $m_a = C_4\|a^*\|_2^2/p$, $M_a = 4(\mathbf{1}^\top a^*)^2 + (3 + C_7/p))\|a^*\|_2^2$ for some constants C_4 and C_7, and

$$T_1 = \widetilde{O}\Big(\frac{p}{\eta}\log\frac{1}{\eta}\log\frac{1}{\delta}\Big).$$

Theorem 6 shows that Phase I of our perturbed GD algorithm only needs polynomial time to ensure the output solution to be sufficiently distant from the spurious local optimum with high probability. Due to the large injected noise, Phase I cannot output a very accurate solution.

Since the proof of Theorem 6 is very technical and involved, we provide a sketch in Appendix C.2, which helps understand the intuition. More details and the proof of all technical lemmas are deferred to C.

As can be seen, a can not make further progress after escaping $\mathcal{A}_{C_2,C_3}$, even when w is more accurate. This is because the injected noise is too large and ruins the accuracy of w. We need decrease the noise level to guarantee convergence.

3.2. Phase II: Converging to the Global Optimum

We then characterize the convergence behavior of the perturbed GD algorithm in Phase II. Recall that in Phase I, the injected noise helps perturbed GD get closer to the global optimum without being trapped in the spurious optimum. Without loss of generality, we restart the iteration index and assume that the initialization (w_0, a_0) follows the result in Theorem 6 :

$$0 < m_a \leq a_0^\top a^* \leq M_a \quad \text{and} \quad \phi_0 \leq \frac{5}{12}\pi,$$

where $m_a = \frac{C_4}{p}\|a^*\|_2^2$ and $M_a = 4(\mathbf{1}^\top a^*)^2 + (3 + \frac{C_7}{p}))\|a^*\|_2^2$.

The next theorem shows that given the reduced injected noise, the perturbed gradient of L with respect to w and a satisfies dissipativity, respectively.

Theorem 7. *For any $\gamma > 0$, we choose $\rho_w^1 \leq C_w^1 \frac{\gamma}{kp} < 1$ and $\rho_a^1 \leq C_a^1$ for small enough constants C_w^1 and C_a^1. Let $\xi \sim \mathrm{unif}(\mathbb{B}_0(\rho_w^1))$ and $\epsilon \sim \mathrm{unif}(\mathbb{B}_0(\rho_a^1))$. For any $C_9 \in (-1, 1]$ and $M > m > 0$, the perturbed manifold gradient of L w.r.t. w satisfies*

$$\begin{aligned}&\langle -\mathbb{E}_{\xi,\epsilon}(I - ww^\top)\nabla_w \mathcal{L}(w+\xi, a+\epsilon), w^* - w\rangle\\ &\qquad\geq \frac{m(1+C_9)}{16}\|w - w^*\|_2^2 - \gamma\end{aligned}$$

for any $(w, a) \in \mathcal{K}_{C_9,m,M}$, where

$$\begin{aligned}\mathcal{K}_{C_9,m,M} = \big\{(w,a)\ \big|\ a^\top a^* \in [m, M],\\ w^\top w^* \geq C_9,\ w \in \mathbb{S}_0(1)\big\}.\end{aligned}$$

Moreover, for any $0 < m < M$ and $C_{10} > 0$, the perturbed gradient of L w.r.t. a satisfies

$$\langle -\mathbb{E}_{\xi,\epsilon}\nabla_a \mathcal{L}(w+\xi, a+\epsilon), a^* - a\rangle \geq \frac{\pi - 1}{2\pi}\|a - a^*\|_2^2 - \gamma$$

for any $(w,a) \in \mathcal{R}_{m,M,C_{10}}$*, where*

$$\mathcal{R}_{m,M,C_{10}} = \big\{(w,a)\big| m \le a^\top a^* \le M, \\ w \in \mathbb{S}_0(1),\ \|w - w^*\|_2^2 \le C_{10}\gamma\big\}.$$

The detailed proof is provided in Appendix D.2. Note that the dissipativity with respect to a depends on the accuracy of w, which indicates that convergence of a happens after that of w. This phenomenon can be seen in the proof of next theorem analyzing the algorithmic behavior in Phase II.

Theorem 8. *Suppose* $\phi_0 \le \frac{5}{12}\pi,\ 0 < m_a \le a_0^\top a^* \le M_a$*. For any* $\gamma > 0$*, we choose* $\rho_w^1 \le C_w^1 \frac{\gamma}{kp} < 1$ *and* $\rho_a \le C_a^1$ *for small enough constants* C_w^1 *and* C_a^1*. For any* $\delta \in (0,1)$*, we choose step size*

$$\eta = C_{11}\Big(\max\Big\{k^4p^6, \frac{k^2p}{\gamma}\Big\}\max\Big\{1, p\log\frac{1}{\gamma}\log\frac{1}{\delta}\Big\}\Big)^{-1}$$

for some constant C_{11}*. Then with at least probability* $1-\delta$*, we have*

$$\|w_t - w^*\|_2^2 \le C_{12}\gamma \quad \textit{and} \quad \|a_t - a^*\|_2^2 \le \gamma$$

for any t*'s such that* $T_2 \le t \le T = \widetilde{O}(\eta^{-2})$*, where* C_{12} *is a constant and*

$$T_2 = \widetilde{O}\Big(\frac{p}{\eta}\log\frac{1}{\gamma}\log\frac{1}{\delta}\Big).$$

Theorem 8 shows that Phase II of our proposed algorithm only needs polynomial time to ensure the convergence to the global optimum with high probability, when the noise is small enough.

Since the proof of Theorem 8 is very technical and involved, we provide a sketch in Appendix D.3, which helps understand the intuition. More details and the proof of all technical lemmas are deferred to D.

4. Extension to Perturbed SGD

Our analysis can be further extended to the perturbed minibatch stochastic gradient descent (perturbed SGD) algorithm. Specifically, we solve

$$\min_{w,a} \mathcal{L}(w,a) \quad \text{subject to} \quad w^\top w = 1,\ \|a\|_2 \le R, \qquad (7)$$

where R is some tuning parameter. At the t-the iteration, we independently sample Gaussian random matrices $Z^{(1)}, ..., Z^{(m)}$, where m is the batch size, and obtains the stochastic approximation of $\nabla\mathcal{L}(w,a)$ by

$$\nabla\widehat{\mathcal{L}}(w,a) = \nabla\left(\frac{1}{m}\sum_{i=1}^m \ell(w,a,Z^{(i)})\right) \\ = \frac{1}{m}\sum_{i=1}^m \nabla\ell(w,a,Z^{(i)}),$$

where $\nabla_w\ell(w,a,Z)$ and $\nabla_a\ell(w,a,Z)$ take the form as follows,

$$\nabla_w\ell(w,a,Z) = \Big(\sum_{j=1}^k a_j a_j^* Z_j Z_j^\top \mathbb{1}_{Z_j^\top w \ge 0, Z_j^\top w^* \ge 0}(w) \\ + \sum_{j \ne i} a_i a_j^* Z_i Z_j^\top \mathbb{1}_{Z_j^\top w \ge 0, Z_j^\top w^* \ge 0}(w)\Big) w^*, \\ \nabla_a\ell(w,a,Z) = \sigma(Zw)\sigma(Zw)^\top a - \sigma(Zw)\sigma(Zw^*)^\top a^*.$$

The perturbed SGD algorithm then takes

$$a_{t+1} = \Pi_{\mathbb{B}_0(R)}\big(a_t - \eta\nabla_a\widehat{\mathcal{L}}(w_t + \xi_t, a_t + \epsilon_t)\big), \\ w_{t+1} = \mathrm{Proj}_{\mathbb{S}_0(1)}\big(w_t - \eta(I - w_t w_t^\top) \\ \cdot \nabla_w\widehat{\mathcal{L}}(w_t + \xi_t, a_t + \epsilon_t)\big),$$

where η is the learning rate, and $\Pi_{\mathbb{B}_0(R)}(\cdot)$ denotes the projection operator to $\mathbb{B}_0(R)$.

Since Z is a Gaussian random matrices with independent entries and w is on the unit sphere, $\sigma(Zw)$ follows a half-normal distribution with variance $(1 - \pi/2)$. Therefore, one can verify that all entries of $Z_j Z_j^\top \mathbb{1}_{Z_j^\top w \ge 0, Z_j^\top w^* \ge 0}(w)$, $Z_i Z_j^\top \mathbb{1}_{Z_j^\top w \ge 0, Z_j^\top w^* \ge 0}(w)$, $\sigma(Zw)\sigma(Zw)^\top$ and $\sigma(Zw)\sigma(Zw^*)^\top$ are sub-exponential random variable with $O(1)$ mean and variance proxy. Due to the space limit, we defer the convergence analysis of perturbed SGD to Appendix F.

5. Numerical Experiment

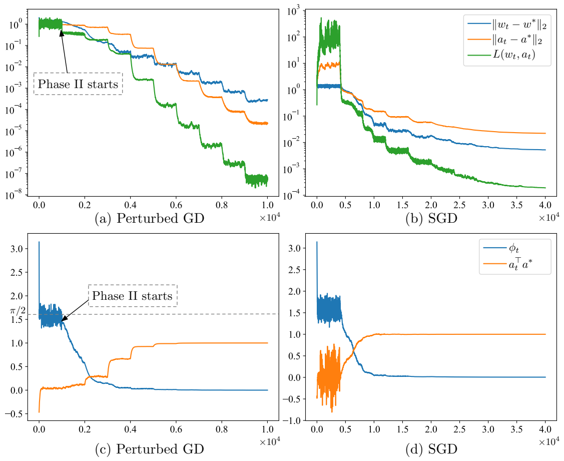
We present numerical experiments to compare our perturbed GD algorithm with GD and SGD.

We first demonstrate that our perturbed GD algorithm with the noise annealing guarantees global convergence to the global optimum. We consider the training of non-overlapping two-layer convolutional neural network in (2) with varying a^* and k. Specifically, we adopt the same experimental setting as in Du et al. (2017). We set $p = 6$ with $k \in \{25, 36, 49, 64, 81, 100\}$ and a^* satisfying

$$\frac{\mathbf{1}^\top a^*}{\|a^*\|_2^2} \in \{0, 1, 4, 9, 16, 25\}.$$

For the perturbed GD algorithm, we perform step size and noise annealing in an epoch-wise fashion: each simulation has 20 epochs with each epoch consisting of 400 iterations; The initial learning rate is 0.1 for both w and a, and geometrically decays with a ratio 0.8; The initial noise levels are given by $(\rho_w, \rho_a) = (36, 1)$ and both geometrically decay with a ratio 0.4. For GD, the learning rate is 0.1 for both w and a. For SGD, we adopt a batch size of 4, and perform step size annealing in an epoch-wise fashion: The initial learning rate is 0.1, and geometrically decays with a ratio 0.4. For perturbed GD and SGD, we purposely initialize at the spurious local optimum. For GD, we adopt the random initialization, as suggested in Du et al. (2017).

For each combination of k and a^*, we repeat 1000 simulations for all three algorithms, and report the success rate of converging to the global optimum in Table 1. As can be seen, perturbed GD and SGD are capable of escaping from the spurious local optimum (even if they are initialized there), and converge to the global optimum throughout all 1000 simulations. However, GD with random initialization can be trapped at the spurious local optimum for up to about 500 simulations. These results are consistent with our theoretical analysis and Du et al. (2017).



(a) Perturbed GD
(b) SGD
(c) Perturbed GD
(d) SGD

Figure 3. Algorithmic Behavior of Perturbed GD and SGD.

We then demonstrate the algorithmic behavior of the perturbed GD algorithm and compare it with SGD. We set $p = 6$, $k = 100$, $a_j^* = -0.1$ for $j = 1, ..., 50$ and $a_j^* = 0.1$ for $j = 51, ..., 100$, and w is randomly generated from the unit sphere. Our selected a^* satisfies $\frac{\mathbf{1}^\top a^*}{\|a^*\|_2^2} = 0$. As suggested by Table 1, this is a difficult case, where GD may get stuck at the spurious local optimum with about 0.5 probability. For the perturbed GD algorithm, we perform step size and noise annealing in an epoch-wise fashion: Each simulation has 10 epochs with each epoch consisting of 1000 iterations; The initial learning rate is 0.1 for both w and a, and geometrically decays with a ratio 0.8; The initial noise levels are given by $(\rho_w, \rho_a) = (36, 1)$ and both geometrically decay with a ratio 0.4. For SGD, we adopt a batch size of 4, and perform step size annealing in an epoch-wise fashion: Each simulation has 10 epochs with each epoch consisting of 4000 iterations; The initial learning rate is 0.1, and geometrically decays with a ratio 0.4. We repeat 10 simulations for both perturbed GD and SGD, and report their (averaged) trajectories in Figure 3.

As can be seen, the trajectories of the perturbed GD algorithm have a phase transition by the end of the second epoch. At the first two epochs, the algorithm makes very slow progress in optimizing a and w due to the large injected noise. Starting from the third epoch, we see that $a_t^\top a^*$ becomes positive and gradually increases, and ϕ_t further decreases. This implies that the algorithm has escaped from the spurious local optimum. Eventually, at later epochs, we see that the algorithm converges to the global optimum, as the magnitude of the injected noise is reduced. These observations are consistent with our theory.

Moreover, we can see that the trajectories of SGD actually show similar patterns to those of the perturbed GD algorithm. At early epochs, only slow progress is made towards optimizing w and a. At later epochs, once SGD escapes from the spurious local optimum, we observe its convergence to the global optimum. Since the noise of SGD comes from the data and has a larger variance than that of the injected noise for the perturbed GD algorithm, we observe more intense oscillation in the trajectories of SGD.

6. Discussions

Partial Dissipativity v.s. Kleinberg et al. (2018). Kleinberg et al. (2018) study the convolutional effect of the noise in nonconvex stochastic optimization, and provide inspiring insights on training neural networks using SGD. Their analysis, however, involves an unconventional asumption. Specifically, they consider a general unconstrained minimization problem $\min_x \mathbb{E}_\xi f(x, \xi)$, and assume

$$\langle -\nabla \mathbb{E}_\xi f(x - \eta \nabla f(x, \xi)),\ x^* - [x - \eta \nabla \mathbb{E}_\xi f(x, \xi)] \rangle \\ \geq c \|x^* - [x - \eta \nabla \mathbb{E}_\xi f(x, \xi)]\|_2^2, \quad (8)$$

where η is the step size of the SGD algorithm. Note that their assumption is essentially imposed over both the optimization problem and the SGD algorithm[4]. However, they do not provide any theoretical evidence showing that such a complicated assumption holds, when applying SGD to any specific nonconvex optimization problem.

The experimental results in Kleinberg et al. (2018) attempt to make some empirical validations of their assumption for training neural networks. Specifically, throughout every iterations of training ResNets and DenseNets, they empirically verify that the following condition holds

$$\left\langle -\frac{1}{m} \sum_{i=1}^{m} \mathbb{E}_\xi \nabla f(x_t + \omega_i, \xi),\ x_t - x^* \right\rangle \geq 0, \quad (9)$$

where ω_i's are independently sampled from a uniform distribution over $\mathbb{B}_0(0.5)$ and $m = 100$. Note that (9) is different from their actual assumption (8).

In contrast, our analysis is dedicated to training two-layer non-overlapping convolutional neural networks in the

[4] The conventional analyses usually impose assumptions on the optimization problem, and all properties of the algorithm need to be proved under the assumptions.

Table 1. Success rates of converging to the global optimum for perturbed GD/GD/SGD with varying k and a^* and $p = 6$.

$\mathbf{1}^\top a^* / \|a^*\|_2^2$	0	1	4	9	16	25
$k = 25$	1.00/0.50/1.00	1.00/0.55/1.00	1.00/0.73/1.00	1.00/1.00/1.00	1.00/1.00/1.00	1.00/1.00/1.00
$k = 36$	1.00/0.50/1.00	1.00 /0.53/1.00	1.00/0.66/1.00	1.00/0.89/1.00	1.00/1.00/1.00	1.00/1.00/1.00
$k = 49$	1.00/0.50/1.00	1.00/0.53/1.00	1.00/0.61/1.00	1.00/0.78/1.00	1.00/1.00/1.00	1.00/1.00/1.00
$k = 64$	1.00/0.50/1.00	1.00/0.51/1.00	1.00/0.59/1.00	1.00/0.71/1.00	1.00/0.89/1.00	1.00/1.00/1.00
$k = 81$	1.00/0.50/1.00	1.00/0.53/1.00	1.00/0.57/1.00	1.00/0.66/1.00	1.00/0.81/1.00	1.00/0.97/1.00
$k = 100$	1.00/0.50/1.00	1.00/0.50/1.00	1.00/0.57/1.00	1.00/0.63/100	1.00/0.75/1.00	1.00/0.90/1.00

teacher/student network setting. The partial dissipative condition used in our analysis can been rigorously verified in Theorems 5 and 7. Moreover, we want to remark that the partial dissipative condition in our analysis is theoretically more challenging, since (1) it does not hold globally; (2) it does not jointly hold over the convolutional weight w and the output weight a; (3) we need to handle the additional errors (e.g., γ_a and γ_w).

Connections to SGD. The motivation of this paper is to understand the role of the noise in training neural network, however, due to the technical limit, directly analyzing SGD is very difficult. The noise of SGD comes from the random sampling of the training data, and it may have a very complex distribution. Moreover, the noise of SGD depends on the iterate, and therefore yields very complicated dependence through iterations. These challenging aspects are beyond our theoretical understanding.

The perturbed GD algorithm considered in this paper is essentially imitating SGD, but easier to analyze: The injected noise follows a uniform distribution and independent on the iterates. Though simpler than SGD, the perturbed GD algorithm has often been observed sharing similar algorithmic behavior to SGD. We remark that from a theoretical perspective, the perturbed GD algorithm is still highly non-trivial and challenging.

Connection to Step Size Annealing. The noise annealing approach is actually closely related to the step size annealing, which has been widely used in training neural networks by SGD. The variance of the noise of SGD has an explicit quadratic dependence on the step size. Therefore, a commonly used practical step size annealing is essentially annealing the noise in training neural networks.

However, we remark that varying step size is actually more complicated than varying noise. When the step size is large, it not only enlarges the noise of SGD, but also encourages aggressive overshooting. This is still beyond our theoretical understanding, as our analysis for the perturbed GD algorithm uses small step sizes with large injected noise.

Algorithmic Behaviors for Training Different Layers. Our analysis shows that the perturbed GD algorithm behaves differently for training the convolutional weight w and the output weight a in Phase I: the algorithm first makes progress in training a, and then makes progress in training w. It is not clear whether this is an artifact of our proof. We believe that some empirical investigations are needed, e.g., examining the training of practical large networks.

Multi-epoch Noise Annealing. Our analysis in Section 3 can be extended to the multi-epoch setting. For instance, we consider a noise level schedule $\{\rho_w^s\}_{s=1}^S$, $\{\rho_a^s\}_{s=1}^S$. When applying our analysis, we can show that there exists a phase transition along the schedule. For the earlier epochs with $\rho_w^s \geq C_w^0 kp^2$ and $\rho_a^s \geq C_a^0$, the algorithm is gradually escaping from the spurious local optimum, which is similar to our analysis for Phase I; For the later epochs with smaller noises, the algorithm is gradually converging to the global optimum, which is similar to our analysis for Phase II.

Overparameterized Neural Networks. Our analysis only considers the regime where the student network has the same architecture as the **teacher** network. This is different from practical situations, where the **student** network is often overparameterized. We conduct some empirical studies on a simple overparameterized case, where the **student** network has two convolutional filters and the **teacher** network has only one convolutional filter. Our studies suggest that such a simple overparameterization does not necessarily lead to a better optimization landscape. There still exist spurious local optima, which can trap the GD algorithm. Due to the space limit, we present the details in Appendix E.

Other Related Works. We briefly discuss several other related works. These works consider different problems, algorithms and assumptions. Therefore, the results are not directly comparable. Specifically, Zhou et al. (2017) study the stochastic mirror descent (*different from ours*) under a global variational coherent assumption (*does not hold for our target problem*); Li & Yuan (2017) study SGD (*different from ours*) for training ResNet-type two-layer neural networks. They assume that the weight of the second layer is known (all one), and prove that the optimization landscape satisfies the one-point convexity over a small neighborhood of the global optimum (*does not hold for our target problem*); Jin et al. (2018) show that the perturbed SGD algorithm for minimizing the empirical risk (*we consider the population risk*), and show that the injected noise rules out the spurious local optima of the empirical risk. However, their assumption requires the population risk to have no spurious local optima (*our population risk contains a spurious local optimum*).

Acknowledgement

This work is partially supported by the grant NSF IIS 1717916 and NSF CMMI-1453934.

References

Barrera, G. and Jara, M. Thermalisation for stochastic small random perturbations of hyperbolic dynamical systems. *arXiv preprint arXiv:1510.09207*, 2015.

Bottou, L. Stochastic gradient learning in neural networks. *Proceedings of Neuro-Nımes*, 91(8):12, 1991.

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

Choromanska, A., Henaff, M., Mathieu, M., Arous, G. B., and LeCun, Y. The loss surfaces of multilayer networks. In *Artificial Intelligence and Statistics*, pp. 192–204, 2015.

Dauphin, Y. N., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., and Bengio, Y. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in neural information processing systems*, pp. 2933–2941, 2014.

Du, S. S., Lee, J. D., Tian, Y., Poczos, B., and Singh, A. Gradient descent learns one-hidden-layer cnn: Don't be afraid of spurious local minima. *arXiv preprint arXiv:1712.00779*, 2017.

Hardt, M. and Ma, T. Identity matters in deep learning. *arXiv preprint arXiv:1611.04231*, 2016.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

Jin, C., Ge, R., Netrapalli, P., Kakade, S. M., and Jordan, M. I. How to escape saddle points efficiently. *arXiv preprint arXiv:1703.00887*, 2017.

Jin, C., Liu, L. T., Ge, R., and Jordan, M. I. On the local minima of the empirical risk. In *Advances in Neural Information Processing Systems*, pp. 4901–4910, 2018.

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

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

Kleinberg, R., Li, Y., and Yuan, Y. An alternative view: When does sgd escape local minima? *arXiv preprint arXiv:1802.06175*, 2018.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Li, Y. and Yuan, Y. Convergence analysis of two-layer neural networks with relu activation. In *Advances in Neural Information Processing Systems*, pp. 597–607, 2017.

Neelakantan, A., Vilnis, L., Le, Q. V., Sutskever, I., Kaiser, L., Kurach, K., and Martens, J. Adding gradient noise improves learning for very deep networks. *arXiv preprint arXiv:1511.06807*, 2015.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning representations by back-propagating errors. *nature*, 323(6088):533, 1986.

Salakhutdinov, R., Mnih, A., and Hinton, G. Restricted boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning*, pp. 791–798. ACM, 2007.

Tian, Y. 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.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

Zhong, K., Song, Z., Jain, P., Bartlett, P. L., and Dhillon, I. S. Recovery guarantees for one-hidden-layer neural networks. *arXiv preprint arXiv:1706.03175*, 2017.

Zhou, Z., Mertikopoulos, P., Bambos, N., Boyd, S., and Glynn, P. W. Stochastic mirror descent in variationally coherent optimization problems. In *Advances in Neural Information Processing Systems*, pp. 7040–7049, 2017.