Local Geometry of Cross Entropy Loss in Learning One-Hidden-Layer Neural Networks

Haoyu Fu*, Yuejie Chi†, Yingbin Liang*
*Dept. of ECE, The Ohio State University, Columbus, OH 43210, USA. Email: {fu.436, liang.889}@osu.edu
†Dept. of ECE, Carnegie Mellon University, Pittsburgh, PA 15213, USA. Email: yuejiechi@cmu.edu

Abstract—**We study model recovery for data classification, where the training labels are generated from a one-hidden-layer neural network with sigmoid activations, and the goal is to recover the weights of the neural network. We consider two network models, the fully-connected network (FCN) and the non-overlapping convolutional neural network (CNN). We prove that with Gaussian inputs, the empirical risk based on cross entropy exhibits strong convexity and smoothness *uniformly* in a local neighborhood of the ground truth, as soon as the sample complexity is sufficiently large. Hence, if initialized in this neighborhood, it establishes the local convergence guarantee for empirical risk minimization using cross entropy via gradient descent for learning one-hidden-layer neural networks, at the near-optimal sample and computational complexity with respect to the network input dimension without unrealistic assumptions such as requiring a fresh set of samples at each iteration.**

I. Introduction

Neural networks have attracted a significant amount of research interest in recent years due to the success of deep neural networks in practical domains such as computer vision and artificial intelligence. However, the theoretical underpinnings behind such success remains mysterious to a large extent. Efforts have been taken to understand which classes of functions can be represented by deep neural networks, when (stochastic) gradient descent is effective for optimizing a nonconvex loss function, and why these networks generalize well.

One important line of research that has attracted extensive attention is the model-recovery problem, which is important for the network to generalize well [1]. Assuming the training samples $(\boldsymbol{x}_i, y_i) \sim (\boldsymbol{x}, y)$, $i = 1, \ldots, n$, are generated independently and identically distributed (i.i.d.) from a distribution $\mathcal{D}$ based on a neural network model with the ground truth parameter $\boldsymbol{W}^\star$, the goal is to recover the underlying model parameter $\boldsymbol{W}^\star$ using the training samples. Consider a network whose output is given as $H(\boldsymbol{W}^\star, \boldsymbol{x})$. Previous studies along this topic can be mainly divided into two cases of data generations, with the input $\boldsymbol{x} \in \mathbb{R}^d$ being Gaussian.

- *Regression*, where each sample $y \in \mathbb{R}$ is generated as
$$y = H(\boldsymbol{W}^\star, \boldsymbol{x}).$$
This type of regression problem has been studied in various settings. In particular, [2] studied the single-neuron model under the Rectified Linear Unit (ReLU) activation, [3] studied the one-hidden-layer multi-neuron network model, and [4] studied a two-layer feedforward network with ReLU activations and identity mapping.
- *Classification*, where the label $y \in \{0, 1\}$ is drawn according to the conditional distribution
$$\mathbb{P}(y = 1|\boldsymbol{x}) = H(\boldsymbol{W}^\star, \boldsymbol{x}).$$
Such a problem has been studied in [5] when the network contains only a single neuron.

For both cases, previous studies attempted to recover $\boldsymbol{W}^\star$, by minimizing an empirical loss function using the squared loss, i.e. $\min_{\boldsymbol{W}} \frac{1}{n}\sum_{i=1}^{n}(y_i - H(\boldsymbol{W}, \boldsymbol{x}_i))^2$, given the training data. Two types of statistical guarantees were provided for such model recovery problems using the squared loss. More specifically, [3] showed that in the local neighborhood of the ground truth $\boldsymbol{W}^\star$, the *empirical* loss function is strongly convex for each *given* point under *independent* high probability event. Hence, their guarantee for gradient descent to converge to the ground truth, assuming proper initialization, requires a *fresh* set of samples at every iteration. Thus the total sample complexity depends on the number of iterations. On the other hand, studies such as [5], [2] established strong convexity in the entire local neighborhood in a uniform sense, so that resampling per iteration is not needed for gradient descent to have guaranteed linear convergence as long as it enters such a local neighborhood. Clearly, the second kind of statistical guarantee *without per-iteration resampling* is much stronger and desirable.

In this paper, we focus on the classification setting by minimizing the empirical loss using the cross entropy objective, which is a popular choice in training practical neural networks. The geometry as well as the model recovery problem based on the cross entropy loss function have not yet been understood even for one-hidden-layer networks. Such a loss function is much more challenging to analyze than the squared loss, not just because it is nonconvex with multiple neurons, but also because its gradient and Hessian take much more complicated forms compared with the squared loss; moreover, it is hard to control the size of gradient and Hessian due to the saturation phenomenon, i.e., when $H(\boldsymbol{W}, \boldsymbol{x})$ approaches 0 or 1. The main focus of this paper is to develop technical analysis for guaranteed model recovery under the challenging cross entropy loss function for the classification problem for two types of one-hidden-layer network structures.

A. Problem Formulation

We consider two popular types of one-hidden-layer nonlinear neural networks, i.e., a Fully-Connected Network (FCN) [3] and a non-overlapping Convolutional Neural Network (CNN) [6]. For both cases, we let $\boldsymbol{x} \in \mathbb{R}^d$ be the input, $K \geq 1$ be the number of neurons, and the activation function be the sigmoid function

$$\phi(x) = \frac{1}{1 + \exp(-x)}.$$

- *FCN:* the network parameter is $\boldsymbol{W} = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_K] \in \mathbb{R}^{d \times K}$, and

$$H_{\text{FCN}}(\boldsymbol{W}, \boldsymbol{x}) = \frac{1}{K} \sum_{k=1}^{K} \phi(\boldsymbol{w}_k^\top \boldsymbol{x}). \tag{1}$$

- *Non-overlapping CNN:* for simplicity we let $d = m \cdot K$ for some integers m. Let $\boldsymbol{w} \in \mathbb{R}^m$ be the network parameter, and the kth stride of $\boldsymbol{x}$ be given as $\boldsymbol{x}^{(k)} = \left[x_{m(k-1)+1}, \cdots x_{m \cdot k}\right]^\top \in \mathbb{R}^m$. Then,

$$H_{\text{CNN}}(\boldsymbol{w}, \boldsymbol{x}) = \frac{1}{K} \sum_{k=1}^{K} \phi(\boldsymbol{w}^\top \boldsymbol{x}^{(k)}). \tag{2}$$

The non-overlapping CNN model can be viewed as a highly structured instance of the FCN, where the weight matrix can be written as:

$$\boldsymbol{W}_{\text{CNN}} = \begin{bmatrix} \boldsymbol{w} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{w} & \dots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \dots & \boldsymbol{w} \end{bmatrix} \in \mathbb{R}^{d \times K}.$$

In a model recovery setting, we are given n training samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n \sim (\boldsymbol{x}, y)$ that are drawn i.i.d. from certain distribution regarding the ground truth network parameter $\boldsymbol{W}^\star$ (or resp. $\boldsymbol{w}^\star$ for CNN). Suppose the network input $\boldsymbol{x} \in \mathbb{R}^d$ is drawn from a standard Gaussian distribution $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$. This assumption has been used a lot in previous literature [2], [7], [6], [8], to name a few. Then, conditioned on $\boldsymbol{x} \in \mathbb{R}^d$, the output y is mapped to $\{0, 1\}$ via the output of the neural network, i.e.,

$$\mathbb{P}(y = 1 | \boldsymbol{x}) = H(\boldsymbol{W}^\star, \boldsymbol{x}). \tag{3}$$

Our goal is to recover the network parameter, i.e., $\boldsymbol{W}^\star$, via minimizing the following empirical loss function:

$$f_n(\boldsymbol{W}) = \frac{1}{n} \sum_{i=1}^{n} \ell(\boldsymbol{W}; \boldsymbol{x}_i), \tag{4}$$

where $\ell(\boldsymbol{W}; \boldsymbol{x}) := \ell(\boldsymbol{W}; \boldsymbol{x}, y)$ is the cross-entropy loss function, i.e.,

$$\ell(\boldsymbol{W}; \boldsymbol{x}) = -y \cdot \log(H(\boldsymbol{W}, \boldsymbol{x})) - (1 - y) \cdot \log(1 - H(\boldsymbol{W}, \boldsymbol{x})), \tag{5}$$

where $H(\boldsymbol{W}, \boldsymbol{x})$ can subsume either H_{FCN} or H_{CNN}.

B. Our Contributions

Considering the multi-neuron classification problem with either FCN or CNN, the main contributions of this work are summarized as follows. Throughout the discussions below, we assume the number K of neurons is a constant, and state the scaling only in terms of the input dimension d and the number n of samples .

- *Uniform local strong convexity:* If the input is Gaussian, the empirical risk function $f_n(\boldsymbol{W})$ is *uniformly* strongly convex in a local neighborhood of the ground truth $\boldsymbol{W}^\star$ as soon as the sample size $n = O(d \log^2 d)$.
- *Statistical and computational rate of gradient descent:* consequently, if initialized in this neighborhood, gradient descent converges linearly to a critical point (which we show to exist). Due to the nature of quantized labels here, the recovery of the ground truth is only up to certain statistical accuracy. In particular, gradient descent finds the critical point $\widehat{\boldsymbol{W}}_n$ with a computation cost of $O(nd \log(1/\epsilon))$, where ϵ denotes the numerical accuracy and $\widehat{\boldsymbol{W}}_n$ converges to $\boldsymbol{W}^\star$ at a rate of $O(\sqrt{d \log n / n})$ in the Frobenius norm.

Note that our performance guarantee of gradient descent requires appropriate initialization. Such an initialization scheme has been provided in an extended version [9], and is omitted due to the space limitations. We derive network specific quantities to capture the local geometry of FCN and CNN, which imply that the geometry of CNN is more benign than FCN, corroborated by the numerical experiments. In order to analyze the challenging cross-entropy loss function, our proof develops various new machineries in order to exploit the statistical information of the geometric curvatures, including the gradient and Hessian of the empirical risk, and to develop covering arguments to guarantee uniform concentrations. To the best of our knowledge, combining the analysis of gradient descent and initialization, this work provides the first globally convergent algorithm for the recovery of one-hidden-layer neural networks using the *cross entropy* loss function.

C. Related Work

Due to the scope, we focus on the most relevant literature on theoretical and algorithmic aspects of learning shallow neural networks via nonconvex optimization.

The studies of one-hidden-layer network model can be further categorized into two classes, landscape analysis and model recovery. In the landscape analysis, it is known that if the network size is large enough compared to the data input, then there are no spurious local minima in the optimization landscape, and all local minima are global [10], [11], [12], [13]. For the case with multiple neurons ($2 \leq K \leq d$) in the under-parameterized setting, there exist spurious bad local minima in the optimization landscape [14], [15] even at the population level. Zhong et. al. [3] provided several important geometric characterizations for the regression problem using a variety of activation functions and the squared loss.

In the model recovery problem, the number of neurons is smaller than the input dimension, and all the existing works discussed below assumed the squared loss and (sub-)Gaussian inputs. [5] showed that when $\phi(\cdot)$ has bounded first, second and third derivatives, there is no other critical points than the unique global minimum (within a constrained region of interest), and (projected) gradient descent converges linearly with an arbitrary initialization, as long as the sample complexity is $O(d\log^2 d)$ for the classification problem. Moreover, in the case with multiple neurons, [7] showed that projected gradient descent with a local initialization converges linearly for smooth activations with bounded second derivatives for the regression problem, [16] showed that gradient descent with tensor initialization converges linearly to a neighborhood of the ground truth using ReLU activations, and [17] showed the linear convergence of gradient descent with the spectral initialization using quadratic activations. For CNN with ReLU activations, [6] showed that gradient descent converges to the ground truth with random initialization for the population risk function based on the squared loss under Gaussian inputs. Moreover, [8] showed that gradient descent learns a two-layer CNN despite the existence of bad local minima. From a technical perspective, our study differs from all the aforementioned work in that the cross entropy loss function we analyze has a very different form. Furthermore, we study the model recovery classification problem under the multi-neuron case, which has not been studied before.

Finally, we note that several papers study one-hidden-layer or two-layer neural networks with different structures under Gaussian input. For example, [18] studied the overlapping convolutional neural network, [4] studied a two-layer feedforward networks with ReLU activations and identity mapping, and [19] introduced the Porcupine Neural Network. Very recently, several papers [20], [21], [22] declared global convergence of gradient descent for optimizing deep neural networks in the over-parameterized regime. These results are not directly comparable to ours since both the networks and the loss functions are different.

D. Notations

Throughout this paper, we use boldface letters to denote vectors and matrices, e.g. $\boldsymbol{w}$ and $\boldsymbol{W}$. The transpose of $\boldsymbol{W}$ is denoted by $\boldsymbol{W}^\top$, and $\|\boldsymbol{W}\|$, $\|\boldsymbol{W}\|_{\mathrm{F}}$ denote the spectral norm and the Frobenius norm. For a positive semidefinite (PSD) matrix $\boldsymbol{A}$, we write $\boldsymbol{A} \succeq 0$. The identity matrix is denoted by $\boldsymbol{I}$. The gradient and the Hessian of a function $f(\boldsymbol{W})$ is denoted by $\nabla f(\boldsymbol{W})$ and $\nabla^2 f(\boldsymbol{W})$, respectively.

We use $c, C, C_1, \ldots$ to denote constants whose values may vary from place to place. For nonnegative functions $f(x)$ and $g(x)$, $f(x) = O\left(g(x)\right)$ means there exist positive constants c and a such that $f(x) \leq cg(x)$ for all $x \geq a$; $f(x) = \Omega\left(g(x)\right)$ means there exist positive constants c and a such that $f(x) \geq cg(x)$ for all $x \geq a$.

II. Gradient Descent and its Performance Guarantee

To estimate the network parameter $\boldsymbol{W}^\star$, since (4) is a highly nonconvex function, vanilla gradient descent with an arbitrary initialization may get stuck at local minima. Therefore, we implement gradient descent (GD) with a well-designed initialization (see the details in [9]). In this section, we focus on the performance of the local update rule

$$\boldsymbol{W}_{t+1} = \boldsymbol{W}_t - \eta \nabla f_n\left(\boldsymbol{W}_t\right),$$

where η is the constant step size. The algorithm is summarized in Algorithm 1.

Algorithm 1 Gradient Descent (GD)

```
Input: Training data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, step size $\eta$, iteration $T$
Initialization: $\boldsymbol{W}_0 \leftarrow$ INITIALIZATION $\left(\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n\right)$
Gradient Descent: for $t = 0, 1, \cdots, T$
    $\boldsymbol{W}_{t+1} = \boldsymbol{W}_t - \eta \nabla f_n\left(\boldsymbol{W}_t\right).$
Output: $\boldsymbol{W}_T$
```

Note that throughout the execution of GD, the same set of training samples is used which is the standard implementation of gradient descent. Consequently the analysis is challenging due to the statistical dependence of the iterates with the data.

A. Geometric Properties of the Networks

Before stating our main results, we first introduce an important quantity $\rho(\sigma)$ regarding $\phi(z)$ that captures the geometric properties of the loss function for neural networks (1) and (2).

Definition 1 (Key quantity for FCN). *Let $z \sim \mathcal{N}\left(0, \sigma^2\right)$ and define $\alpha_q(\sigma) = \mathbb{E}[\phi'(\sigma \cdot z)z^q], \forall q \in \{0,1,2\}$, and $\beta_q(\sigma) = \mathbb{E}[\phi'(\sigma \cdot z)^2 z^q], \forall q \in \{0,2\}$. Define $\rho_{\mathrm{FCN}}(\sigma)$ as*

$$\rho_{\mathrm{FCN}}(\sigma) = \min\left\{\beta_0(\sigma) - \alpha_0^2(\sigma), \beta_2(\sigma) - \alpha_2^2(\sigma)\right\} - \alpha_1^2(\sigma).$$

Definition 2 (Key quantity for CNN). *Let $z \sim \mathcal{N}\left(0, \sigma^2\right)$ and define $\rho_{\mathrm{CNN}}(\sigma)$ as*

$$\rho_{\mathrm{CNN}}(\sigma) = \min\left\{\mathbb{E}[(\phi'(z)z)^2], \mathbb{E}[\phi'\left(z\right)^2]\right\}.$$

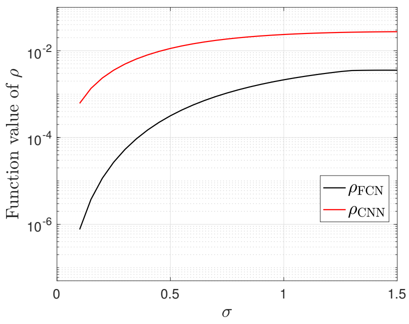

Fig. 1. Illustration $\rho\left(\sigma\right)$ for both FCN and CNN with the sigmoid activation.

Note that Definition 1 for FCN is different from that in [3, Property 3.2] but consistent with [3, Lemma D.4] which removes the third term in [3, Property 3.2]. For the activation function considered in this paper, the first two terms suffice.

Definition 2 for CNN is a newly distilled quantity in this paper tailored to the special structure of CNN. We depict $\rho(\sigma)$ as a function of σ in a certain range for the sigmoid activation in Fig. 1. It can be numerically verified that $\rho(\sigma) > 0$ for all $\sigma > 0$. Furthermore, the value of $\rho_{\mathrm{CNN}}(\sigma)$ is much larger than $\rho_{\mathrm{FCN}}(\sigma)$ for the same input.

B. Uniform Local Strong Convexity

We first characterize the local strong convexity of $f_n(\cdot)$ in a neighborhood of the ground truth. We use the Euclidean ball to denote the local neighborhood of $\boldsymbol{W}^\star$ for FCN or of $\boldsymbol{w}^\star$ for CNN.

$$\mathbb{B}\left(\boldsymbol{W}^\star, r\right) = \left\{\boldsymbol{W} \in \mathbb{R}^{d\times K} : \|\boldsymbol{W} - \boldsymbol{W}^\star\|_{\mathrm{F}} \le r\right\}, \tag{6a}$$

$$\mathbb{B}\left(\boldsymbol{w}^\star, r\right) = \left\{\boldsymbol{w} \in \mathbb{R}^{m} : \|\boldsymbol{w} - \boldsymbol{w}^\star\|_2 \le r\right\}, \tag{6b}$$

where r is the radius of the ball. With slight abuse of notations, we will drop the subscript FCN or CNN for simplicity, whenever it is clear from the context that the result is for FCN when the argument is $\boldsymbol{W} \in \mathbb{R}^{d\times K}$ and for CNN when the argument is $\boldsymbol{w} \in \mathbb{R}^m$. Further, $\sigma_i\left(\boldsymbol{W}\right)$ denotes the i-th singular value of $\boldsymbol{W}^\star$. Let the condition number be $\kappa = \sigma_1/\sigma_K$, and $\lambda = \prod_{i=1}^K \left(\sigma_i/\sigma_K\right)$. The following theorem guarantees the Hessian of the empirical risk function in the local neighborhood of the ground truth is positive definite with high probability for both FCN and CNN.

Theorem 1 (Local Strong Convexity)**.** *Consider the classification model with FCN* (1) *or CNN* (2) *and the sigmoid activation function.*

- *For FCN, assume $\|\boldsymbol{w}_k^\star\|_2 \le 1$ for all k. There exist constants c_1 and c_2 such that as soon as*

$$n_{\mathrm{FCN}} \ge c_1 \cdot dK^5 \log^2 d \cdot \left(\frac{\kappa^2\lambda}{\rho_{\mathrm{FCN}}\left(\sigma_K\right)}\right)^2,$$

  with probability at least $1 - d^{-10}$, we have for all $\boldsymbol{W} \in \mathbb{B}(\boldsymbol{W}^\star, r_{\mathrm{FCN}})$,

$$\Omega\left(\frac{1}{K^2}\cdot\frac{\rho_{\mathrm{FCN}}\left(\sigma_K\right)}{\kappa^2\lambda}\right)\cdot \boldsymbol{I} \preceq \nabla^2 f_n\left(\boldsymbol{W}\right) \preceq \Omega(1)\cdot\boldsymbol{I},$$

  where $r_{\mathrm{FCN}} := \frac{c_2}{\sqrt{K}}\cdot\frac{\rho_{\mathrm{FCN}}(\sigma_K)}{\kappa^2\lambda}$.
- *For CNN, assume $\|\boldsymbol{w}^\star\|_2 \le 1$. There exist constants c_3 and c_4 such that as soon as*

$$n_{\mathrm{CNN}} \ge c_3 \cdot dK^5 \log^2 d \cdot \left(\frac{1}{\rho_{\mathrm{CNN}}\left(\|\boldsymbol{w}^\star\|_2\right)}\right)^2,$$

  with probability at least $1 - d^{-10}$, we have for all $\boldsymbol{w} \in \mathbb{B}(\boldsymbol{w}^\star, r_{\mathrm{CNN}})$,

$$\Omega\left(\frac{1}{K}\cdot\rho_{\mathrm{CNN}}\left(\|\boldsymbol{w}^\star\|_2\right)\right)\cdot \boldsymbol{I} \preceq \nabla^2 f_n\left(\boldsymbol{w}\right) \preceq \Omega(K)\cdot\boldsymbol{I},$$

  where $r_{\mathrm{CNN}} := \frac{c_4}{K^2}\cdot\rho_{\mathrm{CNN}}\left(\|\boldsymbol{w}^\star\|_2\right)$.

We note that for FCN (1), all column permutations of $\boldsymbol{W}^\star$ are equivalent global minimum of the loss function, and Theorem 1 applies to all such permutation matrices of $\boldsymbol{W}^\star$.

Theorem 1 guarantees that for both FCN (1) and CNN (2) the Hessian of the empirical cross-entropy loss function $f_n(\boldsymbol{W})$ is positive definite in a neighborhood of the ground truth $\boldsymbol{W}^\star$, as long as the sample size n is sufficiently large. The bounds in Theorem 1 depend on the dimension parameters of the network (n and K), as well as the ground truth ($\rho_{\mathrm{FCN}}(\sigma_K)$, λ, $\rho_{\mathrm{CNN}}\left(\|\boldsymbol{w}^\star\|_2\right)$).

C. Performance Guarantees of GD

For the classification problem, due to the nature of quantized labels, $\boldsymbol{W}^\star$ is no longer a critical point of $f_n(\boldsymbol{W})$. By the strong convexity of the empirical risk function $f_n(\boldsymbol{W})$ in the local neighborhood of $\boldsymbol{W}^\star$, there can exist at most one critical point in $\mathbb{B}(\boldsymbol{W}^\star, r)$, which is the unique local minimizer in $\mathbb{B}\left(\boldsymbol{W}^\star, r\right)$ if it exists. The following theorem shows that there indeed exists such a critical point $\widehat{\boldsymbol{W}}_n$, which is provably close to the ground truth $\boldsymbol{W}^\star$, and gradient descent converges linearly to $\widehat{\boldsymbol{W}}_n$.

Theorem 2 (Performance Guarantees of Gradient Descent)**.** *Assume the assumptions in Theorem 1 hold. Under the event that local strong convexity holds,*

- *for FCN, there exists a critical point in $\mathbb{B}(\boldsymbol{W}^\star, r_{\mathrm{FCN}})$ such that*

$$\left\|\widehat{\boldsymbol{W}}_n - \boldsymbol{W}^\star\right\|_{\mathrm{F}} \le c_1 \frac{K^{9/4}\kappa^2\lambda}{\rho_{\mathrm{FNN}}\left(\sigma_K\right)}\sqrt{\frac{d\log n}{n}},$$

  and if the initial point $\boldsymbol{W}_0 \in \mathbb{B}(\boldsymbol{W}^\star, r_{\mathrm{FCN}})$, GD converges linearly to $\widehat{\boldsymbol{W}}_n$, i.e.

$$\left\|\boldsymbol{W}_t - \widehat{\boldsymbol{W}}_n\right\|_{\mathrm{F}} \le \left(1 - \frac{c_2\eta\rho_{\mathrm{FCN}}\left(\sigma_K\right)}{K^2\kappa^2\lambda}\right)^t \left\|\boldsymbol{W}_0 - \widehat{\boldsymbol{W}}_n\right\|_{\mathrm{F}},$$

  for $\eta \le c_3$, where c_1, c_2, c_3 are constants;
- *for CNN, there exists a critical point in $\mathbb{B}(\boldsymbol{w}^\star, r_{\mathrm{CNN}})$ such that*

$$\left\|\widehat{\boldsymbol{w}}_n - \boldsymbol{w}^\star\right\|_2 \le c_4 \frac{K}{\rho_{\mathrm{CNN}}\left(\|\boldsymbol{w}^\star\|_2\right)}\cdot\sqrt{\frac{d\log n}{n}},$$

  and if the initial point $\boldsymbol{w}_0 \in \mathbb{B}(\boldsymbol{w}^\star, r_{\mathrm{CNN}})$, GD converges linearly to $\widehat{\boldsymbol{w}}_n$, i.e.

$$\left\|\boldsymbol{w}_t - \widehat{\boldsymbol{w}}_n\right\|_2 \le \left(1 - \frac{c_5\eta\rho_{\mathrm{CNN}}\left(\|\boldsymbol{w}^\star\|_2\right)}{K}\right)^t \left\|\boldsymbol{w}_0 - \widehat{\boldsymbol{w}}_n\right\|_2,$$

  for $\eta \le c_6/K$, where c_4, c_5, c_6 are constants.

Similarly to Theorem 1, for FCN (1) Theorem 2 also holds for all column permutations of $\boldsymbol{W}^\star$. Theorem 2 guarantees that the existence of critical points in the local neighborhood of the ground truth, which GD converges to, and also shows that the critical points converge to the ground truth $\boldsymbol{W}^\star$ at the rate of $O(K^{9/4}\sqrt{d\log n/n})$ for FCN (1) and $O\left(K\sqrt{d\log n/n}\right)$ for CNN(2) with respect to increasing the sample size n. Therefore, $\boldsymbol{W}^\star$ can be recovered consistently as n goes to infinity. Notice that the sample complexity requirement in Theorem 1 depends linearly on the dimension $O(d)$ of the unknown parameters, hence it's near optimal. Moreover, for both FCN (1) and CNN (2) gradient descent converges linearly to $\widehat{\boldsymbol{W}}_n$ (or resp. $\widehat{\boldsymbol{w}}_n$)

at a linear rate, as long as it is initialized in the basin of attraction. To achieve ϵ-accuracy, i.e. $\left\|\boldsymbol{W}_t - \widehat{\boldsymbol{W}}_n\right\|_{\mathrm{F}} \le \epsilon$ (or resp. $\left\|\boldsymbol{w}_t - \widehat{\boldsymbol{w}}_n\right\|_2 \le \epsilon$), it requires a computational complexity of $O\left(ndK^4 \log\left(1/\epsilon\right)\right)$ (or resp. $O\left(ndK^2 \log\left(1/\epsilon\right)\right)$), which is linear in n, d and $\log(1/\epsilon)$.

III. Conclusions

In this paper, we have studied the model recovery problem of a one-hidden-layer neural network using the cross-entropy loss in a multi-neuron classification problem. In particular, we have characterized the sample complexity to guarantee local strong convexity in a neighborhood (whose size we have characterized as well) of the ground truth when the training data are generated from a classification model for two types of neural network models: fully-connected network and non-overlapping convolutional network. This guarantees that with high probability, gradient descent converges linearly to the ground truth if initialized properly. In the future, it will be interesting to extend the analysis in this paper to more general class of activation functions, particularly ReLU-like activations.

Acknowledgment

The work of H. Fu and Y. Liang is supported in part by U.S. National Science Foundation under the grants CCF-1761506 and CCF-1801855. The work of Y. Chi is supported in part by AFOSR under the grant FA9550-15-1-0205, by ONR under the grant N00014-18-1-2142, by ARO under the grant W911NF-18-1-0303, and by NSF under the grants CAREER ECCS-1818571, ECCS-1833553, and CCF-1806154.

Appendix A
Proof Sketch

We sketch the proof of both Theorem 1 and Theorem 2 in this section, and the complete proof can be found in an extended version online [9].

In order to show that the empirical loss possesses a local strong convexity, we follow the following steps:

1) We first show that the Hessian $\nabla^2 f(\boldsymbol{W})$ of the population loss function is smooth with respect to $\nabla^2 f(\boldsymbol{W}^\star)$;
2) We then show that $\nabla^2 f(\boldsymbol{W})$ satisfies local strong convexity and smoothness in a neighborhood of $\boldsymbol{W}^\star$ with appropriately chosen radius, $\mathbb{B}(\boldsymbol{W}^\star, r)$, by leveraging similar properties of $\nabla^2 f(\boldsymbol{W}^\star)$;
3) Next, we show that the Hessian of the empirical loss function $\nabla^2 f_n(\boldsymbol{W})$ is close to its population counterpart $\nabla^2 f(\boldsymbol{W})$ uniformly in $\mathbb{B}(\boldsymbol{W}^\star, r)$ with high probability.
4) Finally, putting all the arguments together, we establish $\nabla^2 f_n(\boldsymbol{W})$ satisfies local strong convexity and smoothness in $\mathbb{B}(\boldsymbol{W}^\star, r)$.

We have established that $f_n\left(\boldsymbol{W}\right)$ is strongly convex in $\mathbb{B}(\boldsymbol{W}^\star, r)$ in Theorem 1. Thus there exists at most one critical point in $\mathbb{B}(\boldsymbol{W}^\star, r)$. The proof of Theorem 2 follows the steps below:

1) We first show that the gradient $\nabla f_n\left(\boldsymbol{W}\right)$ concentrates around $\nabla f\left(\boldsymbol{W}\right)$ in $\mathbb{B}(\boldsymbol{W}^\star, r)$, and then invoke [5, Theorem 2] to guarantee that there indeed exists a critical point $\widehat{\boldsymbol{W}}_n$ in $\mathbb{B}(\boldsymbol{W}^\star, r)$;
2) We next show that $\widehat{\boldsymbol{W}}_n$ is close to $\boldsymbol{W}^\star$ and gradient descent converges linearly to $\widehat{\boldsymbol{W}}_n$ with a properly chosen step size.

References

[1] M. Mondelli and A. Montanari, "On the connection between learning two-layers neural networks and tensor decomposition," *arXiv preprint arXiv:1802.07301*, 2018.

[2] M. Soltanolkotabi, "Learning relus via gradient descent," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 2007–2017.

[3] K. Zhong, Z. Song, P. Jain, P. L. Bartlett, and I. S. Dhillon, "Recovery guarantees for one-hidden-layer neural networks," in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, 2017, pp. 4140–4149.

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

[5] S. Mei, Y. Bai, and A. Montanari, "The landscape of empirical risk for nonconvex losses," *Ann. Statist.*, vol. 46, no. 6A, pp. 2747–2774, 12 2018.

[6] A. Brutzkus and A. Globerson, "Globally optimal gradient descent for a ConvNet with Gaussian inputs," in *Proceedings of the 34th International Conference on Machine Learning*, 2017, pp. 605–614.

[7] S. Oymak, "Learning compact neural networks with regularization," in *Proceedings of the 35th International Conference on Machine Learning*, 2018, pp. 3966–3975.

[8] S. S. Du, J. D. Lee, and Y. Tian, "When is a convolutional filter easy to learn?" in *International Conference on Learning Representations*, 2018.

[9] H. Fu, Y. Chi, and Y. Liang, "Local geometry of one-hidden-layer neural networks for logistic regression," *arXiv preprint arXiv:1802.06463*, 2018.

[10] M. Soltanolkotabi, A. Javanmard, and J. D. Lee, "Theoretical insights into the optimization landscape of over-parameterized shallow neural networks," *IEEE Transactions on Information Theory*, 2018.

[11] D. Boob and G. Lan, "Theoretical properties of the global optimizer of two layer neural network," *arXiv preprint arXiv:1710.11241*, 2017.

[12] I. Safran and O. Shamir, "On the quality of the initial basin in overspecified neural networks," in *International Conference on Machine Learning*, 2016, pp. 774–782.

[13] Q. Nguyen and M. Hein, "The loss surface of deep and wide neural networks," in *International Conference on Machine Learning*, 2017, pp. 2603–2612.

[14] R. Ge, J. D. Lee, and T. Ma, "Learning one-hidden-layer neural networks with landscape design," in *International Conference on Learning Representations*, 2018.

[15] I. Safran and O. Shamir, "Spurious local minima are common in two-layer ReLU neural networks," in *Proceedings of the 35th International Conference on Machine Learning*, 2018, pp. 4433–4441.

[16] X. Zhang, Y. Yu, L. Wang, and Q. Gu, "Learning one-hidden-layer relu networks via gradient descent," *arXiv preprint arXiv:1806.07808*, 2018.

[17] Y. Li, C. Ma, Y. Chen, and Y. Chi, "Nonconvex matrix factorization from rank-one measurements," *arXiv preprint arXiv:1802.06286*, 2018.

[18] S. Goel, A. Klivans, and R. Meka, "Learning one convolutional layer with overlapping patches," in *Proceedings of the 35th International Conference on Machine Learning*, 2018, pp. 1783–1791.

[19] S. Feizi, H. Javadi, J. Zhang, and D. Tse, "Porcupine neural networks: Approximating neural network landscapes," in *Advances in Neural Information Processing Systems 31*, 2018, pp. 4836–4846.

[20] Z. Allen-Zhu, Y. Li, and Z. Song, "A convergence theory for deep learning via over-parameterization," *arXiv preprint arXiv:1811.03962*, 2018.

[21] S. S. Du, J. D. Lee, H. Li, L. Wang, and X. Zhai, "Gradient descent finds global minima of deep neural networks," *arXiv preprint arXiv:1811.03804*, 2018.

[22] D. Zou, Y. Cao, D. Zhou, and Q. Gu, "Stochastic gradient descent optimizes over-parameterized deep relu networks," *arXiv preprint arXiv:1811.08888*, 2018.