Spurious Local Minima Exist for Almost All Over-parameterized Neural Networks

Tian Ding* Dawei Li † Ruoyu Sun ‡

Oct 4, 2019

Abstract

A popular belief for explaining the efficiency in training deep neural networks is that over-paramenterized neural networks have nice landscape. However, it still remains unclear whether over-parameterized neural networks contain spurious local minima in general, since all current positive results cannot prove non-existence of bad local minima, and all current negative results have strong restrictions to the activation functions, data samples or network architecture. In this paper we answer this question with a surprisingly negative result. In particular, we prove that for almost all deep over-parameterized non-linear neural networks, spurious local minima exist for generic input data samples. Our result helps give a more exact characterization of the landscape of deep neural networks and corrects a long-believed misunderstanding in the past decades.

1 Introduction

It is widely believed that over-parameterized neural networks have nice landscape [1, 2, 3, 4, 5]. Recent theoretical works [6, 7, 8, 9] proved "GD can converge to global minima" for deep neural networks under the assumptions of a huge number of neurons per layer and special initialization. These works are essentially local analysis in a quite small neighborhood of the global minima, and

*Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong. dt016@ie.cuhk.edu.hk. The work is done while the author is visiting Department of ISE, Univeristy of Illinois at Urbana-Champaign. The author contributes equally to this paper.

†Coordinated Science Laboratory, Department of ISE, University of Illinois at Urbana-Champaign, Urbana, IL. dawei2@illinois.edu. The author contributes equally to this paper.

‡Coordinated Science Laboratory, Department of ISE, University of Illinois at Urbana-Champaign, Urbana, IL. ruoyus@illinois.edu.

it is not clear whether practical training is restricted to that small neighborhood. Instead of local analysis, one might ask whether a clean global analysis can be achieved for neural networks under reasonable conditions. In particular, a natural question to ask is: *Do (over-parameterized) neural networks contain spurious local minima?*

For linear networks, people have gained a rich understanding and are convinced that the answer is "yes". Baldi and Hornik [10] proved that no bad local minimum exists for shallow networks, and Kawaguchi [11] first proved a similar result for deep linear networks under mild assumptions. A series of works provided proofs under milder conditions [12, 13, 14].

For nonlinear networks, perhaps quite surprisingly, there are very few results that really prove "no spurious local minima". Yu et al. [15] proved that 1-hidden-layer over-parameterized nonlinear networks has nice landscape property for generic data. Since then, lots of works have focused on over-parameterization and some of them have proved non-existence of spurious valley [16, 17, 18] or bad basins [19]. Nevertheless, none of them have proved the elimination of all spurious local minima. In fact, the only paper that claimed that no bad local minimum exists is Yu et al. [15], but it was found by Li et al. [19] that their proof had a cavity and the claim does not hold. Our paper aims to understand whether the lack of "no bad local-min" result is due to intrinsic barrier, or the limitation of technical skills.

1.1 What Conditions Shall We Impose

We impose very mild assumptions on the network width, the activations and the data. Below, we explain that if one adds extra assumptions on these components, then bad local minima can be constructed.

Wide Network (Over-parameterized). The classical work Auer et al.[20] presented a concrete counter-example where exponentially many spurious local minima exist in a single-neuron network. However, the counter-example was an unrealizable case (i.e. the network cannot fit data), and the authors proved that under the same setting, bad local minima would not exist if the network can fit data. Therefore, it is of little interest to show bad local minima exist for unrealizable cases. An ideal counter-example would be for the realizable case; in addition, it will be even better if it is for the wide-network setting where one layer can have more neurons than the number of samples.

Smooth Activations. Due to the popularity of ReLU activations, a few works showed that ReLU networks have bad local minima (e.g., Swirszcz et al.[21] Zhou et al. [22], Safran et al.[23], Venturi et al.[16], Liang et al.[24][1]). One

[1] Safran et al.[23] and Venturi et al.[16] both provided counter-examples when the objective

intuition why ReLU can lead to bad local minima is that it can create flat regions ("dead regions") which are bad local minima. Intuitively, such flat regions may disappear if the activations are smoooth (this is indeed proved in Liang et al.[24] for special data). Indeed, the positive results [15, 16, 17, 18] are all for smooth activations, which seem to indicate that smooth activations have better landscape than non-smooth activations. Therefore, an ideal counter-example should apply to smooth activations.

Generic Data. There are a few works (Liang et al. [24] and Yun et al.[25]) that construct bad local minima for smooth activations, but in their examples the data points lie in a zero-measure space. [2] However, the existing positive results, including the classical work [15] and recent works [16, 17, 18] all assume generic data [3]. Thus an ideal counter-example should apply to generic data, or at least a positive measure of data points.

In summary, all the existing counter-examples are restricted since they assume special data or non-smooth activations. These special assumptions can lead to the existence of spurious local minima, but it is still possible that simple (and practical) changes can eliminate bad local minima. In particular, for deep and wide neural networks with smooth activations and generic data (the common settting in previous results), it seems possible that no bad local minima exist. In fact, previous results already eliminate bad basins under this setting, and only allow the existence of flat local minima. It seems we are only a tiny step away from a clean result. Perhaps surprisingly, we will show this seemingly small gap is insurmountable (without extra assumptions).

1.2 Our Contributions

We consider a supervised learning problem where the prediction is parameterized by a multi-layer neural network. We prove that for any fully connected neural networks (arbitrarily wide), a large class of smooth activations (dense in the set of continuous functions) and generic input data samples, there exists an example that spurious local minima exist. Our examples are much broader than the previous constructions of spurious local minima which rely on special components.

In detail, our contribution includes (suppose d is the input dimension, and n is the number of samples):

---

function is the population risk (a different setting from the empirical risk minimization).

[2] These works have extra restrictions. Liang et al. [24] only considers activations that satisfy $\sigma(t) + \sigma(-t) = c, \forall t$. Yun et al.[25] only considers a network with two-neurons and three data points and thus not a wide-network setting.

[3] More rigorously, a result holds for "generic" data means that except for a zero-measure set, the result holds.

- We show that for almost all analytic activation functions, and all realizable fully-connected neural networks, if $d < O(\sqrt{n})$, then we can find the corresponding output data samples such that spurious local minima exist. This implies that under the setting of [15, 16, 17, 18, 19], it is impossible to prove "no spurious local-min" without additional assumptions.
- We show that for all realizable deep neural networks with activation that contains a linear segment, spurious local minima exist for generic training samples.
- We adopt a simple but effective approach for the construction of spurious local minima. The proof technique is novel and can be generalized to broader settings.

To our knowledge, our result is the first counter-example result that covers generic non-linear neural networks and generic input data samples. More importantly, this result reveals the fact that "generic deep neural networks do have spurious local minima", which corrects a long-believed misunderstanding in the past decades. It is noteworthy that our result does not imply "neural networks have bad landscape". Instead, many works have shown that neural networks have landscape that is slightly inferior to "no bad local minima"[18, 19]. Together with these works, our work gives a complete characterization of the landscape neural networks possess.

We summarize the existing exmaples and our examples in Table 1

Table 1: Summary of existing examples and our examples

Reference	Width	Realizable	Activation	Data
Auer et al.	1	No	Generic[2]	Fixed
Swirszcz et al.	2 or 3	No	Sigmoid, ReLU	Fixed
Zhou et al.	1	No	ReLU	Fixed
Safran et al.[1]	6 to 20	Yes	ReLU	Gaussian
Venturi et al.[1]	Any	No	$L^2(\mathbb{R}, e^{-x^2/2})$	Adversarial
Liang et al.	Any	Yes	$\sigma(t) + \sigma(-t) = c$	Fixed
Yun et al.	2	Yes	Small nonlinearity	Fixed
This paper	Any	Yes	Generic nonlinearity	Generic input

[1] In these two examples, the objective function is the population risk, which is a different setting from the empirical risk minimization.

[2] The actual requirement is that $l(\cdot, \sigma(\cdot))$ is continuous and bounded, where $l(\cdot)$ and $\sigma(\cdot)$ are the loss function and the activation function separately.

The paper is organized as follows. We first present the network model we study in Section 2. In Section 3, we present the main results. The main proof idea is provided in Section 4 with an example proof for the 1-hidden-layer-1-input-

dimensional case. Then a complete proof is presented in Section 5. We finally make some discussions and conclusions in Section 6.

2 Network Model

2.1 Network Structure

Consider a fully connected neural network with H hidden layers. Assume that the h-th hidden layer contains d_h neurons for $1 \leq h \leq H$, and the input and output layers contain d_0 and d_{H+1} neurons, respectively. Given an input sample $\mathbf{x} \in \mathbb{C}^{d_0}$, the input of the i-th neuron of the h-th hidden layer, denoted by $z_{i,j}$, is given by

$$z_{1,i}(\mathbf{x}) = \sum_{j=1}^{d_0} w_{1,i,j}\mathbf{x}_j + b_{1,i}, \quad 1 \leq i \leq d_1 \tag{1a}$$

$$z_{h,i}(\mathbf{x}) = \sum_{j=1}^{d_{h-1}} w_{h,i,j}t_{h-1,j}(\mathbf{x}) + b_{h,i}, \quad 1 \leq i \leq d_h, \quad 2 \leq h \leq H \tag{1b}$$

where x_j is the j-th entry of the input data, $w_{h,i,j}$ is the weight from the j-th neuron of the $(h-1)$-th layer to the i-th neuron of the h-th layer, $b_{h,i}$ is the bias added to the i-th neuron of the h-th layer. Let σ be the neuron activation function. Then the output of the i-th neuron of the h-th hidden layer, denoted by $t_{h,i}$, is given by

$$t_{h,i}(\mathbf{x}) = \sigma\left(z_{h,i}(\mathbf{x})\right), \quad 1 \leq i \leq d_h, \quad 1 \leq h \leq H. \tag{2}$$

Finally, the i-th output of the network, denoted by $t_{H+1,i}$, is given by

$$t_{H+1,i}(\mathbf{x}) = \sum_{j=1}^{d_H} w_{H+1,i,j}t_{H,i}(\mathbf{x}), \quad 1 \leq i \leq d_{H+1} \tag{3}$$

where $w_{H+1,i,j}$ is the weight to the output layer, defined similarly to that of the hidden layers.

Then, we define $W_h \in \mathbb{R}^{d_{h-1} \times d_h}$ as the weight matrix from the $(h-1)$-th layer to the h-th layer, and $\mathbf{b}_h \in \mathbb{R}^{d_h}$ as the bias vector of the h-th layer. The entries of each matrix are given by

$$(W_h)_{i,j} = w_{h,i,j}, \quad (\mathbf{b}_h)_i = b_{h,i}, \tag{4}$$

2.2 Training Data

Consider a training dataset consisting of N samples. Noting that the input dimension and the output dimension are both one, we denote the n-th sample by $(x^{(n)}, y^{(n)})$, $n = 1, \cdots, N$, where $x^{(n)}, y^{(n)} \in \mathbb{R}$ are the input and output samples, respectively. We can rewrite all the samples in vector forms, i.e.

$$X \triangleq [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N] \in \mathbb{R}^{1\times N} \tag{5a}$$

$$Y \triangleq [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_N] \in \mathbb{R}^{1\times N}. \tag{5b}$$

With the input data given, we can represent the input and output of each hidden-layer neuron by

$$z_{h,i,n} = z_{h,i}(\mathbf{x}_n) \tag{6a}$$

$$t_{h,i,n} = t_{h,i}(\mathbf{x}_n) \tag{6b}$$

for $h = 1, 2, \cdots, H$, $i = 1, 2, \cdots, d_h$, and $n = 1, 2, \cdots, N$. Then, we define $Z_h \in \mathbb{R}^{d_h\times N}$ and $T_h \in \mathbb{R}^{d_h\times N}$ as the input and output matrix of the h-th layer with

$$(Z_h)_{n,i} = z_{h,i,n} \tag{7a}$$

$$(T_h)_{n,i} = t_{h,i,n}. \tag{7b}$$

Similarly, we denote the output matrix by $\hat{Y} \in \mathbb{C}^{d_{H+1}\times N}$, where

$$(\hat{Y})_{i,n} = \hat{y}_{i,n} = t_{H+1,i}(\mathbf{x}_n) \tag{8a}$$

for $i = 1, 2, \cdots, d_{H+1}$, $n = 1, 2, \cdots, N$

2.3 Training Loss

Let W denote all the network weights, i.e.

$$W = (W_1, \mathbf{b}_1, W_2, \mathbf{b}_2 \cdots, W_H, \mathbf{b}_H, W_{H+1}) \tag{9}$$

In this notes, we consider the quadratic loss function to characterize the training error. That is, given the training dataset (X, Y), the empirical loss is given by

$$E(W) = ||Y - \hat{Y}(W)||_F^2. \tag{10}$$

Here we treat the network output $\hat{Y}$ as a function of all the weights. Then, the training problem of the considered network is to find W to minimize the empirical loss $E(W)$.

3 Main Theorems

In this subsection, we present our main results. Before we present the theorems, we first specify our assumptions on the input dataset and the activation functions.

Assumption 1

a) The input dimension d_0 satisfies $d_0^2 + d_0 < N$.

b) The following $d_0^2 + d_0 + 1$ vectors

$$\mathcal{X} = \{\mathbf{1}, X_{(1,:)}, X_{(2,:)}, \cdots, X_{(N,:)}, X_{(1,:)} \circ X_{(1,:)}, X_{(1,:)} \circ X_{(2,:)}, \cdots, X_{(i,:)} \circ X_{(j,:)}, \cdots, X_{(d_0,:)} \circ X_{(d_0,:)}\} \quad (11)$$

are linearly independent. Note that $\mathcal{X}$ includes all the rows of X and the Hadamard product between any two rows of X.

Assumption 2 *There exists $a \in \mathbb{R}$ and $\delta > 0$ such that*

a) σ is twice differentiable on $[a - \delta, a + \delta]$.

b) $\sigma(a), \sigma'(a), \sigma''(a) \neq 0$.

Now we are able to present our first main theorem.

Theorem 1 *Consider a fully-connected deep neural network with $H \geq 2$. Suppose that Assumption 1 and 2 hold. Then there exits $Y \in \mathbb{R}^{d_{H+1} \times N}$ such that the empirical loss $E(W)$ has local minimum W with $E(W) > 0$.*

We next consider activation functions which is linear in at least a small interval.

Assumption 3

a) There exists $a \in \mathbb{R}$ and $\delta > 0$, such that σ is linear in $(a - \delta, a + \delta)$.

b) Each hidden layer is wider than the input layer, i.e., $d_h > d_0$ for $h = 1, 2, \cdots, H$.

c) The training data (X, Y) satisfies $\mathrm{rank}([X^\top, \mathbf{1}, Y^\top]) > \mathrm{rank}([X^\top, \mathbf{1}])$.

Theorem 2 *Consider a fully-connected deep neural network with $H \geq 2$. Suppose that Assumption 3 holds. Then the empirical loss $E(W)$ has local minimum W with $E(W) > 0$.*

4 Proof Idea

In this section, we use a 1-hidden-layer example to demonstrate the key idea in finding bad local minima in over-parameterized neural networks. In particular, we prove the following theorem.

Theorem 3 *Suppose we are given $N \geq 3$ input data samples $x_1, \cdots, x_N \in \mathbb{R}$. Consider a 1-hidden-layer neural network with $d \geq N$ neurons. The neural network can be represented as $\hat{y}^\top = v^\top \sigma(wx^\top)$ where $v, w \in \mathbb{R}^d$. Assume that*

- *The input data samples $x_i, i = 1, \cdots, N$ are distinguished;*
- *The loss function is quadratic, i.e. $l(y, \hat{y}) = \|y - \hat{y}\|_2^2$;*
- *The activation function $\sigma(\cdot)$ satisfies Assumption A2.*

Then there exists N output data samples $y_i, i = 1, \cdots, N$ such that the objective function $E(w, v)$ has bad non-strict local minima.

To prove this theorem, we take the following steps:

Step 1: Decomposing the difference of empirical loss.

Consider an arbitrarily perturbation from (v, w) to $(v', w') = (v + \Delta v, w + \Delta w)$. Assume that $\hat{y}' = v'^\top \sigma(w' x^\top)$, then the objective function after perturbation is given by $E(v', w') = l(y, \hat{y}') = \|y - \hat{y}'\|^2$. Therefore, the difference of the objective function before and after perturbation is

$$E(v', w') - E(v, w) = \|y - \hat{y}'\|^2 - \|y - \hat{y}\|^2 = 2(y - \hat{y})^\top (\hat{y} - \hat{y}') + \|\hat{y} - \hat{y}'\|^2.$$

Note that $\|\hat{y} - \hat{y}'\|^2$ is always non-negative, to make $E(v', w') - E(v, w) \geq 0$, it is sufficient to prove $\langle y - \hat{y}, \hat{y}' - \hat{y} \rangle \leq 0$. Since we can arbitrarily choose y, from now on we only need to handle with $\hat{y}' - \hat{y}$.

Step 2: Expand $\hat{y}' - \hat{y}$ into second-order term.

Note that

$$\begin{aligned}(\hat{y}' - \hat{y})^\top &= (v + \Delta v)^\top \sigma((w + \Delta w)x^\top) - v^\top \sigma(wx^\top) \\ &= \sum_{i=1}^{d} (v_i + \Delta v_i)\sigma((w_i + \Delta w_i)x^\top) - \sum_{i=1}^{d} v_i \sigma(w_i x^\top)\end{aligned}$$

By Taylor expansion, we can rewrite $\sigma((w_i + \Delta w_i)x^\top)$ as

$$\begin{aligned}\sigma((w_i + \Delta w_i)x^\top) =& (\sigma((w_i + \Delta w_i)x_1), \cdots, \sigma((w_i + \Delta w_i)x_N))^\top \\ =& z_i^\top + \partial z_i^\top + \partial^2 z_i^\top + o(\Delta w_i)^2 \cdot \mathbf{1}^\top, \quad 1 \leq i \leq d\end{aligned}$$

where $\mathbf{1}$ is the all-1 vector, and

$$\begin{aligned} z_i &= (\sigma(w_i x_1), \cdots, \sigma(w_i x_N))^\top, \quad 1 \le i \le d, \\ \partial z_i &= (x_1 \sigma'(w_i x_1), \cdots, x_N \sigma'(w_i x_N))^\top, \quad 1 \le i \le d, \\ \partial^2 z_i &= (\frac{1}{2} x_1^2 \sigma''(w_i x_1), \cdots, \frac{1}{2} x_N^2 \sigma''(w_i x_N))^\top, \quad 1 \le i \le d. \end{aligned}$$

where $\sigma'(\cdot)$ and $\sigma''(\cdot)$ are the first and second derivative of $\sigma(\cdot)$ respectively.

Thus we can represent $\hat{y}' - \hat{y}$ as

$$\begin{aligned} \hat{y}' - \hat{y} &= \sum_{i=1}^{d} (v_i + \Delta v_i)(z_i + \partial z_i + \partial^2 z_i + o(\Delta w_i)^2 \cdot \mathbf{1}) - \sum_{i=1}^{d} v_i z_i \\ &= \sum_{i=1}^{d} (\Delta v_i z_i + v_i \Delta w_i \partial z_i) + \sum_{i=1}^{d} (\Delta v_i \Delta w_i \partial z_i + v_i (\Delta w_i)^2 \partial^2 z_i) + o(\Delta w_i)^2 \cdot \mathbf{1} \end{aligned}$$

For simplicity, denote $y^p = y - \hat{y}$. Combining Step 1 and Step 2, we rewrite the desired inequality $\langle y - \hat{y}, \hat{y}' - \hat{y} \rangle \le 0$ as

$$\begin{aligned} 0 &\ge \langle y - \hat{y}, \hat{y}' - \hat{y} \rangle \\ &= \sum_{i=1}^{d} \Delta v_i \langle y^p, z_i \rangle + \sum_{i=1}^{d} (\Delta v_i \Delta w_i + v_i \Delta w_i) \langle v^p, \partial z_i \rangle \\ &+ \sum_{i=1}^{d} v_i (\Delta w_i)^2 \langle y^p, \partial^2 z_i \rangle + o(\Delta w_i)^2 \cdot \langle y^p, \mathbf{1} \rangle \end{aligned} \tag{12}$$

Step 3: Solve a linear system to satisfy equation (12).

The final step is to select proper w^*, v^* and y^* such that equation (12) holds. Note that in (12), the sign of the second-order term is not related to Δv or Δw. Therefore, we can make $\langle y^p, z_i \rangle = \langle v^p, \partial z_i \rangle = 0$ and $v_i \cdot \langle y^p, \partial^2 z_i \rangle < 0$ for $i = 1, \cdots, d$ so that the non-positive terms dominate the right hand side of (12).

Specifically, let $w^* = \mathbf{0}$, then $z_1 = \cdots = z_d$, and the right hand side of (12) becomes

$$\begin{aligned} &\left(\sum_{i=1}^{d} \Delta v_i \right) \cdot \langle y^p, z_1 \rangle + \left(\sum_{i=1}^{d} (\Delta v_i \Delta w_i + v_i \Delta w_i) \right) \cdot \langle v^p, \partial z_1 \rangle \\ &+ \left(\sum_{i=1}^{d} v_i (\Delta w_i)^2 \right) \cdot \langle y^p, \partial^2 z_1 \rangle + o(\Delta w_i)^2 \cdot \langle y^p, \mathbf{1} \rangle. \end{aligned} \tag{13}$$

To this end, we introduce a very simple lemma.

Lemma 1 *Given $z_1, z_2, z_3 \in \mathbb{R}^N$ where $N \geq 3$. Assume that z_3 is not a linear combination of z_1 and z_2, then there exists $y \in \mathbb{R}^N$ such that $y^\top z_1 = y^\top z_2 = 0$ and $y^\top z_3 \neq 0$.*

Proof:[Proof of Lemma 1] Let $W = \text{span}\{z_1, z_2\}$. Decompose z_3 into $z_3 = u+v$, where $u \in W$ and $v \in W^\perp$. Since z_3 is not a linear combination of z_1 and z_2, $v \neq \mathbf{0}$, so $v^\top z_3 \neq 0$. Moreover, $v^\top z_1 = v^\top z_2 = 0$. Taking $y = v$ completes the proof. □

Since Assumption A2 holds, the first 3-by-3 submatrix of $\begin{bmatrix} z_1^\top \\ \partial z_1^\top \\ \partial^2 z_1^\top \end{bmatrix}$,which has the form

$$\begin{bmatrix} \sigma(0) & \sigma(0) & \sigma(0) \\ x_1\sigma'(0) & x_2\sigma'(0) & x_3\sigma'(0) \\ \frac{1}{2}x_1^2\sigma''(0) & \frac{1}{2}x_2^2\sigma''(0) & \frac{1}{2}x_3^2\sigma''(0) \end{bmatrix},$$

is a Vandermonde matrix with each row scaled by a non-zero constant. Thus, $\partial^2 z_1$ is not a linear combination of z_1 and ∂z_1. According to Lemma 1, there exists $(y^p)^* \in \mathbb{R}^N$ such that $\langle (y^p)^*, z_i \rangle = \langle (y^p)^*, \partial z_i \rangle = 0$ and $\langle (y^p)^*, \partial^2 z_i \rangle \neq 0$. Let $y^* = \hat{y} + (y^p)^*$ and $v_i^* = -\text{sgn}\langle (y^p)^*, \partial^2 z_1 \rangle$ for all $1 \leq i \leq d$. Now expression (13) turns into

$$\left(\sum_{i=1}^{d} v_i (\Delta w_i)^2 \right) \cdot \langle y^p, \partial^2 z_1 \rangle + o(\Delta w_i)^2 \cdot \langle y^p, \mathbf{1} \rangle. \tag{14}$$

If $\Delta w_i = 0$ for all $i = 1, \cdots, d$, then (14) is constant 0. If there exists some $1 \leq i \leq d$ such that $\Delta w_i \neq 0$, $\left(\sum_{i=1}^{d} v_i (\Delta w_i)^2 \right) \cdot \langle y^p, \partial^2 z_1 \rangle \leq (\Delta w_i)^2 \cdot v_i \langle y^p, \partial^2 z_1 \rangle$ is strictly negative. Moreover, it dominates $o(\Delta w_i)^2 \cdot \langle y^p, \mathbf{1} \rangle$ for sufficently small Δw_i. Therefore, (14) is always non-positive, which implies that (12) always holds. So we have shown that (w^*, v^*) is a bad local minimum when the output samples are selected as y^*. The proof is complete.

We provide some concluding remarks about this proof. Seeing through the proof procedure, what is actually done is expressing the difference of the empirical loss into a second-order Taylor expansion. After removing some quadratic terms, we find that the remaining terms have simple expression. In particular, the signs of the second-order terms are easy to control despite the existence of perturbation. Therefore, we control the sign of the remaining terms by zeroing out the zero-order and first-order terms so that the second-order terms dominate the whole expression. Specifically, the zeroing-out process is achieved by solving linear systems.

Although deep neural networks seem to have much more complicated expressions, the same procedure can be utilized.

5 Proof of Theorem 1

5.1 Preliminaries

For convenience, we first introduce the following notations. For $1 \leq h_1 \leq h_2 \leq H$, let

$$W_{[h_1:h_2]} = (W_{h_1}, \mathbf{b}_{h_1}, W_{h_1+1}, \mathbf{b}_{h_1+1}, \cdots, W_{h_2}, \mathbf{b}_{h_2}) \tag{15}$$

be the weights from the h_1-th layer to the h_2-th layer and

$$W_{[h_1:(H+1)]} = (W_{h_1}, \mathbf{b}_{h_1}, W_{h_1+1}, \mathbf{b}_{h_1+1}, \cdots, W_H, \mathbf{b}_H, W_{H+1}) \tag{16}$$

be the weights from the h_1-th layer to the $(H+1)$-th layer. Then for the i-th neuron in the h-th hidden layer, the input and output is a function of $W_{[1:h]}$ and $\mathbf{x}_n$, written as $t_{h,i}(W_{[1:h]}, \mathbf{x}_n)$ and $z_{h,i}(W_{[1:h]}, \mathbf{x}_n)$, respectively.

For two weight settings W and W', we denote

$$\tilde{W}' = \left(W_1, \mathbf{b}_1', W_2', \mathbf{b}_2', \cdots, W_H', \mathbf{b}_H', W_{H+1}'\right) \tag{17}$$

where the weights to the first hidden layer are picked from W, while the bias to the first hidden layer and the remaining weights and bias are all from W'.

5.2 Local Minimum Construction

We construct the weights as follows.

(1) $W_1 = \mathbf{0}$;

(2) $w_{h,i,j} > 0, \quad h = 2, \cdots, H, H+1 \quad i = 1, \cdots, d_i, \quad j = 1, \cdots, d_{i-1}$;

(3) $\mathbf{b}_1 = a \cdot \mathbf{1}$;

(4) $b_{h,i} = a - \sigma(a) \sum_{k=1}^{d_{h-1}} w_{h,i,j}, \quad h = 2, \cdots, H, \quad i = 1, \cdots, d_h$.

We would like to make some comments on the construction above.

First, we see that in (1), the weights to the first hidden layer are set to be zero, and in (2) the weights to other hidden layers are arbitrary values with the same sign as $\sigma'(a)$, and the weights to all other layers are arbitrary positive values. This implies that there exits $\delta_1 > 0$ such that for any $W' \in B(W, \delta_1)$, conditions (2) are also satisfied by W', i.e.

$$w_{h,i,j}' > 0, \quad h = 2, \cdots, H+1, \quad \forall i, j \tag{18a}$$

Second, It can be readily verified that with bias satisfying (3) and (4), for any input sample the input to all hidden-layer neurons is a, so we have $t_{h,i,n} = \sigma(a)$ for all h, i, n. Notice that σ is twice differentiable on $[a-\delta, a+\delta]$. Therefore there exists $\delta_2 > 0$ such that for any $W' \in B(W, \delta_2)$, the input of each hidden neuron is within $(a-\delta, a+\delta)$ and the sign of the output does not change, i.e.

$$z_{h,i}(W') \in (a-\delta, a+\delta) \tag{19a}$$

$$t_{h,i}(W') * \sigma(a) > 0 \tag{19b}$$

for $h = 1, \cdots, H$, and $i = 1, \cdots, d_h$. Then, within $B(W, \delta_2)$, the input and output of each neuron are twice differentiable functions with respect to the weights.

In the remaining proof, whenever we consider a weight perturbation W' around W, we always assume $W' \in B(W, \min\{\delta_1, \delta_2\})$.

Now, let $\hat{Y}(W)$ be the resulting network output of the constructed weights. We then pick the training output data Y such that each row of $\Delta Y \triangleq \hat{Y}(W) - Y$ satisfies

$$\langle \Delta Y_{(i,:)}, \mathbf{1}^\top \rangle = 0 \tag{20a}$$

$$\langle \Delta Y_{(i,:)}, X_{(j,:)} \rangle = 0 \tag{20b}$$

$$\langle \Delta Y_{(i,:)}, X_{(j,:)} \circ X_{(j',:)} \rangle = 0 \tag{20c}$$

$$[\sigma'(a)]^{H-1} \sigma''(a) \langle \Delta Y_{(i,:)}, X_{(j,:)} \circ X_{(j,:)} \rangle > 0 \tag{20d}$$

For any $i = 1, 2, \cdots, d_{H+1}$ and $j, j' = 1, 2, \cdots, d_0$ with $j \neq j'$.

To guarantee the existence of such Y, we present the following lemma.

Lemma 2 *Consider a fully-connected deep neural network with $H \geq 2$. Suppose that Assumption 1 hold. Then for any W, there exists Y satisfying* (20).

To prove Theorem 1, what remains is to show that for the constructed W and Y, W is a local minimum of the empirical loss with $E(W) > 0$.

5.3 Perturbation Direction

Consider a small perturbation W' around the constructed W. The resulting difference of the training loss is given by

$$\begin{aligned} & E(W') - E(W) \\ =& ||\hat{Y}(W') - Y||_F^2 - ||\hat{Y}(W) - Y||_F^2 \\ =& 2\langle \Delta Y, \hat{Y}(W') - \hat{Y}(W) \rangle_F + ||\hat{Y}(W') - \hat{Y}(W)||_F^2 \end{aligned} \tag{21}$$

Therefore $E(W') - E(W) \geq 0$ if

$$\langle \Delta Y, \hat{Y}(W') - \hat{Y}(W) \rangle_F \geq 0. \tag{22}$$

We can further decompose $\hat{Y}(W') - \hat{Y}(W)$ as

$$\hat{Y}(W') - \hat{Y}(W) = \hat{Y}(\tilde{W}') - \hat{Y}(W) + \hat{Y}(W') - \hat{Y}(\tilde{W}') \tag{23}$$

To prove that W is a local minimum, it suffices to show that for any W' that is sufficiently close to W, we have

$$\langle \Delta Y, \hat{Y}(\tilde{W}') - \hat{Y}(W) \rangle_F \geq 0 \tag{24a}$$

$$\langle \Delta Y, \hat{Y}(\tilde{W}') - \hat{Y}(W) \rangle_F \geq 0 \tag{24b}$$

We first show that, for the constructed W and any W', (24a) holds.

In fact, if $W_1 = \mathbf{0}$, each network output $t_{H+1,i}(\mathbf{x})$ is invariant to the input vector $\mathbf{x}$. Therefore, we have

$$\hat{y}_{i,1}(W) = \hat{y}_{i,2}(W) = \cdots = \hat{y}_{i,N}(W) \tag{25a}$$

$$\hat{y}_{i,1}(\tilde{W}') = \hat{y}_{i,2}(\tilde{W}') = \cdots = \hat{y}_{i,N}(\tilde{W}') \tag{25b}$$

for $i = 1, 2, \cdots, d_{H+1}$. Thus, for W and $\tilde{W}'$, each row of the network output matrix can be written as

$$\hat{Y}_{(i,:)}(W) = \hat{y}_{i,1}(W) \cdot \mathbf{1}^\top \tag{26a}$$

$$\hat{Y}_{(i,:)}(\tilde{W}') = \hat{y}_{i,1}(\tilde{W}') \cdot \mathbf{1}^\top \tag{26b}$$

$$\tag{26c}$$

and from (20a) we have

$$\begin{aligned} &\langle \Delta Y, \hat{Y}(\tilde{W}') - \hat{Y}(W) \rangle_F \\ =& \sum_{i=1}^{d_{H+1}} \langle \Delta Y_{(i,:)}, \hat{Y}_{(i,:)}(\tilde{W}') - \hat{Y}_{(i,:)}(W) \rangle \\ =& \sum_{i=1}^{d_H+1} \Big[y_{i,1}(\tilde{W}') - \hat{y}_{i,1}(W) \Big] \cdot \langle \Delta Y_{(i,:)}, \mathbf{1}^\top \rangle = 0, \end{aligned} \tag{27}$$

implying that (24a) is satisfied.

Then we present the following lemma.

Lemma 3 *Consider a fully-connected deep neural network with $H \geq 2$. Suppose that Assumption 1 and 2 hold. Then for the W and Y constructed in Section 5.2, there exists $\delta_3 > 0$ such that for any $W' \in B(W, \delta_3)$*

$$\langle \Delta Y, \hat{Y}(W') - \hat{Y}(\tilde{W}') \rangle_F \geq 0 \tag{28}$$

where the equality holds if and only if $||W_1' - W_1||_F^2 = 0$

Therefore, (24b) is satisfied by W' that is sufficiently close to W. We complete the proof.

6 Conclusion

In this paper, we studied the existence of spurious local minima in deep nonlinear neural networks. Specifically, for deep networks with almost all analytic activations, we show that bad local minima exist if the dimension of the input data sample is smaller than the square root of the number of data samples. For deep networks with activations that contain a linear segment, we prove that bad local minima exist for generic training samples without any assumption. Our result solves a long-standing question of "whether spurious local minima exist in general deep neural networks", and the answer is somewhat astonishingly negative. Nevertheless, combining with other positive results, we believe that this work reveals the exact landscape of deep neural networks, which is not as nice as people generally think but much better than general non-convex functions. This work also provides a future research direction of how to avoid such spurious local minima effectively in a general setting during the training process, and calls for a deeper understanding of the empirical efficiency of training deep neural networks.

References

[1] Yoshua Bengio, Nicolas L Roux, Pascal Vincent, Olivier Delalleau, and Patrice Marcotte. Convex neural networks. In *NIPS*, pages 123–130, 2006.

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

[3] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. *ICLR*, 2017.

[4] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems*, pages 5947–5956, 2017.

[5] Mario Geiger, Stefano Spigler, Stéphane d'Ascoli, Levent Sagun, Marco Baity-Jesi, Giulio Biroli, and Matthieu Wyart. The jamming transition as a paradigm to understand the loss landscape of deep neural networks. *arXiv preprint arXiv:1809.09349*, 2018.

[6] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. *arXiv preprint arXiv:1811.03962*, 2018.

[7] Simon S Du, Jason D Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. *arXiv preprint arXiv:1811.03804*, 2018.

[8] Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu. Stochastic gradient descent optimizes over-parameterized deep ReLU networks. *arXiv preprint arXiv:1811.08888*, 2018.

[9] Difan Zou and Quanquan Gu. An improved analysis of training over-parameterized deep neural networks. *arXiv preprint arXiv:1906.04688*, 2019.

[10] P. Baldi and K. Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, 2(1):53–58, 1989.

[11] K. Kawaguchi. Deep learning without poor local minima. In *NIPS*, pages 586–594, 2016.

[12] Haihao Lu and Kenji Kawaguchi. Depth creates no bad local minima. *arXiv preprint arXiv:1702.08580*, 2017.

[13] Thomas Laurent and James Brecht. Deep linear networks with arbitrary loss: All local minima are global. In *International Conference on Machine Learning*, pages 2908–2913, 2018.

[14] Li Zhang. Depth creates no more spurious local minima. *arXiv preprint arXiv:1901.09827*, 2019.

[15] Xiao-Hu Yu and Guo-An Chen. On the local minima free condition of back-propagation learning. *IEEE Transactions on Neural Networks*, 6(5):1300–1303, 1995.

[16] Luca Venturi, Afonso Bandeira, and Joan Bruna. Spurious valleys in two-layer neural network optimization landscapes. *arXiv preprint arXiv:1802.06384*, 2018.

[17] Quynh Nguyen, Mahesh Chandra Mukkamala, and Matthias Hein. On the loss landscape of a class of deep neural networks with no bad local valleys. *arXiv preprint arXiv:1809.10749*, 2018.

[18] Quynh Nguyen. On connected sublevel sets in deep learning. *arXiv preprint arXiv:1901.07417*, 2019.

[19] Dawei Li, Tian Ding, and Ruoyu Sun. Over-parameterized deep neural networks have no strict local minima for any continuous activations. *arXiv preprint arXiv:1812.11039*, 2018.

[20] Peter Auer, Mark Herbster, and Manfred K Warmuth. Exponentially many local minima for single neurons. In *Advances in neural information processing systems*, pages 316–322, 1996.

[21] Grzegorz Swirszcz, Wojciech Marian Czarnecki, and Razvan Pascanu. Local minima in training of deep networks. 2016.

[22] Yi Zhou and Yingbin Liang. Critical points of neural networks: Analytical forms and landscape properties. *arXiv preprint arXiv:1710.11205*, 2017.

[23] Itay Safran and Ohad Shamir. Spurious local minima are common in two-layer relu neural networks. *ICML*, 2018.

[24] S. Liang, R. Sun, Y. Li, and R. Srikant. Understanding the loss surface of neural networks for binary classification. 2018.

[25] Chulhee Yun, Suvrit Sra, and Ali Jadbabaie. Small nonlinearities in activation functions create bad local minima in neural networks. *arXiv preprint arXiv:1802.03487*, 2018.

A Proof of Theorem 2

From Assumption 3, σ is linear in $(a-\delta, a+\delta)$, say

$$\sigma(t) = \alpha t + \beta, \quad t \in (a-\delta, a+\delta). \tag{29}$$

Now we construct the weights to each hidden layer such that the following two conditions are satisfied.

(1) $z_{h,i,n} \in (a-\delta, a+\delta)$, $\forall i, n$;

(2) $\text{row}\,(T_h(W)) = \text{row}\left(\begin{bmatrix} X \\ \mathbf{1}_N^\top \end{bmatrix}\right)$.

Consider the weights to the first hidden layer. Notice that $d_1 > d_0$, we let

$$W_1 = V_1 \in \mathbb{R}^{d_1 \times d_0} \tag{30}$$

where $V_1 \in \mathbb{R}^{d_1 \times d_0}$ satisfies $||V_1 X||_F^2 < \delta/2$, to be determined later. Let $\mathbf{b}_1 = a\mathbf{1}_{d_1} + \mathbf{u}_1$, where $\mathbf{u}_1 \in \mathbb{R}^N$ satisfies $||\mathbf{u}_1||_F^2 < \delta/2$, also to be determined later. Then we can verify that condition (1) holds for the first hidden layer. We further

have

$$\begin{aligned} T_1 =&\sigma\left([W_1, \mathbf{b}_1]\begin{bmatrix} X \\ \mathbf{1}_N^\top \end{bmatrix}\right) \\ =&\alpha W_1 X + \alpha \mathbf{b}\mathbf{1}_N^\top + \beta \mathbf{1}_{d_1 \times N} \\ =&\alpha V_1 X + \left[\alpha \mathbf{u}_1 + (\alpha a + \beta)\mathbf{1}_{d_1}\right]\mathbf{1}_N^\top \\ =&\left[\alpha V_1, \alpha \mathbf{u}_1 + (\alpha a + \beta)\mathbf{1}_{d_1}\right]\begin{bmatrix} X \\ \mathbf{1}_N^\top \end{bmatrix} \end{aligned} \tag{31}$$

There exist V_1 and $\mathbf{u}_1$ with $||V_1 X||_F, ||\mathbf{u}||_2 < \delta/2$, such that

$$\text{row}(T_1) = \text{row}\left(\begin{bmatrix} X \\ \mathbf{1}_N^\top \end{bmatrix}\right). \tag{32}$$

Thus, condition (2) is also satisfied. If conditions (1) and (2) hold for the $(h-1)$-th hidden layer, following a similar analysis, we can construct W_h and $\mathbf{b}_h$ to meet conditions (1) and (2) for the h-th hidden layer. As such, we construct $W_{[1:H]}$. Finally, we consider the weights in the output layer, i.e., W_{H+1}. We let

$$W_{H+1} \in \arg\min_{V \in \mathbb{R}^{d_{H+1} \times d_H}} ||Y - V T_H||_F^2. \tag{33}$$

Note that condition (2) holds for the last hidden layer, and therefore W equivalently minimizes the distance from Y to row $\left(\begin{bmatrix} X \\ \mathbf{1}_N^\top \end{bmatrix}\right)$, i.e.,

$$E(W) = \min_{V \in \mathbb{R}^{d_{H+1} \times (d_H+1)}} \left|\left|Y - V\begin{bmatrix} X \\ \mathbf{1}_N^\top \end{bmatrix}\right|\right|_F^2. \tag{34}$$

From Assumption 3, $E(W) > 0$.

To complete the proof, it suffices to show that the constructed W is indeed a local minimum. From assumption 3, there exists δ_1 such that for any $W' \in B(W, \delta_1)$, the input of any hidden-layer neuron is within $(a - \delta, a + \delta)$. Then, it can show that for $h = 1, 2, \cdots, H$,

$$\text{row}(T_h(W')) \in \text{row}\left(\begin{bmatrix} X \\ \mathbf{1}_N^\top \end{bmatrix}\right). \tag{35}$$

Therefore,

$$\begin{aligned} E(W') =& \left|\left|Y - W'_{H+1} T_H\left(W'_{[1:H]}\right)\right|\right|_F^2 \\ \geq& \min_{V \in \mathbb{R}^{d_{H+1} \times (d_H+1)}} \left|\left|Y - V\begin{bmatrix} X \\ \mathbf{1}_N^\top \end{bmatrix}\right|\right|_F^2 \\ =& E(W) \end{aligned} \tag{36}$$

Thus, W is a local minimum with $E(W) > 0$.

B Proof of Lemma 2

Without loss of generality, we assume $[\sigma'(a)]^{H-1}\sigma''(a) > 0$.

We first construct an $N \times d_0$ matrix $X^{(1)}$ whose columns consist of the transpose of all vectors in

$$\mathcal{X}_1 = \{X_{(i,:)} \circ X_{(i,:)} | i = 1, 2, \cdots, d_0\} \tag{37}$$

which is a subset of $\mathcal{X}$, and an $N \times \left(d_0^2 + 1\right)$ matrix $X^{(2)}$ whose columns consist of the transpose of all vectors in $\mathcal{X} \backslash \mathcal{X}_1$.

As the vectors in $\mathcal{X}$ are linearly independent, $X^{(1)}$ and $X^{(2)}$ are both full column rank, i.e., $\text{rank}(X^{(1)}) = d_0$ and $\text{rank}(X^{(2)}) = d_0^2 + 1$. Further, we have

$$\text{rank}\left(\left[X^{(1)}, X^{(2)}\right]\right) = d_0^2 + d_0 + 1. \tag{38}$$

This implies that there exists $V \in \mathbf{R}^{d_0 \times N}$ such that

$$VX^{(1)} = I, \quad VX^{(2)} = \mathbf{0}. \tag{39}$$

Now we construct each row of Y as

$$Y_{(i,:)} = \hat{Y}_{(i,:)}(W) - \sum_{j=1}^{d_0} \alpha_{i,j} V_{(j,:)}, \quad i = 1, 2, \cdots, d_{H+1} \tag{40}$$

where each $\alpha_{i,j}$ is an arbitrary positive value. Then, from (39) we have

$$\begin{aligned} &[\sigma'(a)]^{H-1}\sigma''(a) \cdot \langle \Delta Y_{(i,:)}, \mathbf{u}_1 \rangle \\ =&[\sigma'(a)]^{H-1}\sigma''(a) \sum_{j=1}^{d_0} \alpha_{i,j} \langle V_{(j,:)}, \mathbf{u}_1 \rangle \\ =&[\sigma'(a)]^{H-1}\sigma''(a) \sum_{j=1}^{d_0} \alpha_{i,j} > 0 \end{aligned} \tag{41}$$

for any $\mathbf{u}_1 \in \mathcal{X}_1$. Thus, (20d) is met. We also have

$$\langle \Delta Y_{(i,:)}, \mathbf{u}_2 \rangle = \sum_{j=1}^{d_0} \alpha_{i,j} \langle V_{(j,:)}, \mathbf{u}_2 \rangle = 0 \tag{42}$$

for any $\mathbf{u}_1 \in \mathcal{X}_2$. Thus, (20a)-(20c) are met. We complete the proof.

C Proof of Lemma 3

First, we show that for each hidden layer, we have the following claim.

Claim 1: For the h-th hidden layer, $h = 1, 2, \cdots, H$, there exists $\delta'_{h,i}$ such that for any $W' \in B(W, \delta'_{h,i})$ with $W'_1 \neq W_1$,

$$[\sigma'(a)]^{(H-h)} \cdot \langle \Delta Y_{(i,:)}, (T_h)_{(j,:)}(W') - (T_h)_{(j,:)}(\tilde{W}') \rangle > 0 \tag{43}$$

for $i = 1, 2, \cdots, d_{H+1}$, $j = 1, 2, \cdots, d_h$.

We prove Claim 1 by induction. Noting that $W_1 = 0$, for the first hidden layer, we have

$$\begin{aligned}
&(T_1)_{(j,:)}(W') - (T_1)_{(j,:)}(\tilde{W}') \\
=&\sigma((W'_1)_{(j,:)}X) - \sigma(a)\mathbf{1}^\top && \text{(44a)} \\
=&\sigma'(a) \cdot (W'_1)_{(j,:)}X + \sigma''(a) \cdot \left[(W'_1)_{(j,:)}X\right] \circ \left[(W'_1)_{(j,:)}X\right] \\
&+ \mathbf{o}(||(W'_1)_{(j,:)}||_2^2) && \text{(44b)} \\
=&\sigma'(a) \cdot (W'_1)_{(j,:)}X + \sigma''(a) \left[\sum_{k=1}^{d_0} w'_{1,j,k} X_{(k,:)}\right] \circ \left[\sum_{k=1}^{d_0} w'_{1,j,k} X_{(k,:)}\right] \\
&+ \mathbf{o}(||(W'_1)_{(j,:)}X||_2^2) && \text{(44c)}
\end{aligned}$$

where (44a) follows from $W_1 = \mathbf{0}$ and (44b) is by Taylor expansion at W_1. From (20b), we have

$$\langle \Delta Y_{(i,:)}, \sigma'(a)(W'_1)_{(j,:)}X \rangle = \sigma'(a) \sum_{k=1}^{d_0} w_{1,j,k} \langle \Delta Y_{(i,:)}, X_{(j,:)} \rangle = 0 \tag{45}$$

and from (20c), we have

$$\begin{aligned}
&\left\langle \Delta Y_{(i,:)}, \sigma''(a) \left[\sum_{k=1}^{d_0} w'_{1,j,k} X_{(k,:)}\right] \circ \left[\sum_{k=1}^{d_0} w'_{1,j,k} X_{(k,:)}\right] \right\rangle \\
=&\sigma''(a) \sum_{k=1}^{d_0} (w'_{1,j,k})^2 \langle \Delta Y_{(i,:)}, X_{(k,:)} \circ X_{(k,:)} \rangle \\
&+ 2\sigma''(a) \sum_{k=1}^{d_0} \sum_{k'=1}^{k-1} w'_{1,j,k} w'_{1,j,k'} \langle \Delta Y_{(i,:)}, X_{(k,:)} \circ X_{(k',:)} \rangle \\
=&\sigma''(a) \sum_{k=1}^{d_0} (w'_{1,j,k})^2 \langle \Delta Y_{(i,:)}, X_{(k,:)} \circ X_{(k,:)} \rangle && \text{(46)}
\end{aligned}$$

Then from (44c), each of the inner products $\langle \Delta Y_{(i,:)}, X_{(k,:)} \circ X_{(k,:)} \rangle$ has the same sign with $[\sigma'(a)]^{H-1}\sigma''(a)$. Further, since $W'_1 \neq \mathbf{0}$, there exists at least

one $w'_{1,j,k} > 0$. Then, we have

$$\begin{aligned}
&[\sigma'(a)]^{H-1} \left\langle \Delta Y_{(i,:)}, \sigma''(a) \left[\sum_{k=1}^{d_0} w'_{1,j,k} X_{(k,:)}\right] \circ \left[\sum_{k=1}^{d_0} w'_{1,j,k} X_{(k,:)}\right] \right\rangle \\
=&\sigma''(a)[\sigma'(a)]^{H-1} \sum_{k=1}^{d_0} (w'_{1,j,k})^2 \langle \Delta Y_{(i,:)}, X_{(k,:)} \circ X_{(k,:)} \rangle \\
>&0
\end{aligned} \tag{47}$$

Finally, notice that there exists $\delta'_{1,i} > 0$ such that for any $W' \in B(W, \delta'_{1,i})$ with $W'_1 \neq W_1$, we have

$$\begin{aligned}
&|\langle \Delta Y_{(i,:)}, \mathbf{o}(||(W'_1)_{(j,:)} X||_2^2) \rangle| \\
\leq&\frac{1}{2} \left| \sigma''(a) \sum_{k=1}^{d_0} (w'_{1,j,k})^2 \langle \Delta Y_{(i,:)}, X_{(k,:)} \circ X_{(k,:)} \rangle \right|.
\end{aligned} \tag{48}$$

Therefore,

$$\begin{aligned}
&[\sigma'(a)]^{H-1} \langle \Delta Y_{(i,:)}, (T_h)_{(j,:)}(W') - (T_h)_{(j,:)}(\tilde{W}') \rangle \\
\geq&\frac{1}{2} [\sigma'(a)]^{H-1} \sigma''(a) \sum_{k=1}^{d_0} (w'_{1,j,k})^2 \langle \Delta Y_{(i,:)}, X_{(k,:)} \circ X_{(k,:)} \rangle \\
>&0
\end{aligned} \tag{49}$$

Now, consider an arbitrary $2 \leq h \leq H$, and suppose that Claim 1 holds for the $(h-1)$-th hidden layer.

$$\begin{aligned}
&(T_h)_{(j,:)}(W') - (T_h)_{(j,:)}(\tilde{W}') \\
=&\sigma((W'_h)_{(j,:)} T_{h-1}(W')) - \sigma((W'_h)_{(j,:)} T_{h-1}(\tilde{W}')) \\
=&\sigma'(a) \cdot (W'_h)_{(j,:)} \left[T_{h-1}(W') - T_{h-1}(\tilde{W}') \right] \\
&+ \mathbf{o}\left(\left|\left| (W'_h)_{(j,:)} \left[T_{h-1}(W') - T_{h-1}(\tilde{W}') \right] \right|\right|_2 \right)
\end{aligned} \tag{50}$$

Noting that each $w'_{h,j,k}$ is positive, there exists $\delta'_{h,i}$ such that for any $W' \in B(W, \delta'_{h,i})$ with $W'_1 \neq W_1$, $||T_{h-1}(W') - T_{h-1}(\tilde{W}')||_2$ is sufficiently small such that

$$\begin{aligned}
&[\sigma'(a)]^{(H-h)} \cdot \langle \Delta Y_{(i,:)}, (T_h)_{(j,:)}(W') - (T_h)_{(j,:)}(\tilde{W}') \rangle \\
>&\frac{1}{2} [\sigma'(a)]^{(H-h+1)} \sum_{k=1}^{d_{h-1}} w'_{h,j,k} \langle \Delta Y_{(i,:)}, (T_{h-1})_{(k,:)}(W') - (T_{h-1})_{(k,:)}(\tilde{W}') \rangle \\
>&0
\end{aligned} \tag{51}$$

We complete the proof of Claim 1.

For the output layer, we have the following claim. Note that based on Claim 1, Claim 2 can be shown in the same way with (50) and (51). We omit the detailed proof of Claim 2 here.

Claim 2: There exists $\delta_3 > 0$ such that for any $W' \in B(W, \delta_3)$ with $W_1' \neq W_1$,

$$\langle \Delta Y, \hat{Y}(W') - \hat{Y}(\tilde{W}') \rangle_F > 0. \tag{52}$$

At last, for any $W' \in B(W, \delta_3)$ with $W_1' = W_1$, we have $W' = \tilde{W}'$. Thus

$$\langle \Delta Y, \hat{Y}(W') - \hat{Y}(\tilde{W}') \rangle_F = \langle \Delta Y, \mathbf{0} \rangle_F = 0. \tag{53}$$

Combining Claim 2, we complete the proof of Lemma 3.