Understanding the Loss Surface of Neural Networks for Binary Classification

Shiyu Liang [1] **Ruoyu Sun** [1] **Yixuan Li** [2] **R. Srikant** [1]

Abstract

It is widely conjectured that training algorithms for neural networks are successful because all local minima lead to similar performance; for example, see (LeCun et al., 2015; Choromanska et al., 2015; Dauphin et al., 2014). Performance is typically measured in terms of two metrics: training performance and generalization performance. Here we focus on the training performance of neural networks for binary classification, and provide conditions under which the training error is zero at all local minima of appropriately chosen surrogate loss functions. Our conditions are roughly in the following form: the neurons have to be increasing and strictly convex, the neural network should either be single-layered or is multi-layered with a shortcut-like connection, and the surrogate loss function should be a smooth version of hinge loss. We also provide counterexamples to show that, when these conditions are relaxed, the result may not hold.

1. Introduction

Local search algorithms like stochastic gradient descent (Bottou, 2010) or variants have gained huge success in training deep neural networks (see, Krizhevsky et al. 2012; Goodfellow et al. 2013; Wan et al. 2013, for example). Despite the spurious saddle points and local minima on the loss surface (Dauphin et al., 2014), it has been widely conjectured that all local minima of the empirical loss lead to similar training performance (LeCun et al., 2015; Choromanska et al., 2015). For example, Li et al. (2015) empirically showed that neural networks with identical architectures but different initialization points can converge to local minima with similar classification performance. However, it still remains a challenge to characterize the theoretical properties of the loss surface for neural networks.

In the setting of regression problems, there have been many recent attempts to study the landscape and/or the training performance of local search algorithms. For shallow models, (Andoni et al., 2014; Sedghi & Anandkumar, 2014; Janzamin et al., 2015; Haeffele & Vidal, 2015; Gautier et al., 2016; Brutzkus & Globerson, 2017; Soltanolkotabi, 2017; Soudry & Hoffer, 2017; Goel & Klivans, 2017; Du et al., 2017; Zhong et al., 2017; Li & Yuan, 2017) provide conditions under which the local search algorithms are guaranteed to converge to the globally optimal solution for the regression problem. For deep linear networks, it has been shown that every local minimum of the empirical loss is a global minimum (Baldi & Hornik, 1989; Kawaguchi, 2016; Freeman & Bruna, 2016; Hardt & Ma, 2017; Yun et al., 2017). In order to characterize the loss surface of nonlinear deep networks for regression tasks, Choromanska et al. (2015) have related the loss function to a spin glass model under a few unrealistic assumptions, and it remains a concern to properly justify their assumptions. More recently, it has been shown (Nguyen & Hein, 2017a;b) that if one layer in the multilayer network has more neurons than the number of training samples, then a subset of local minima are global minima.

Although the loss surfaces in regression tasks have been extensively studied, the theoretical understanding of loss surfaces in classification tasks is still limited. (Nguyen & Hein, 2017b; Boob & Lan, 2017; Soltanolkotabi et al., 2017) treat the classification problem as the regression problem by using quadratic loss. However, the global minimum of the quadratic loss does not necessarily have zero misclassification error even in the simplest cases (e.g., even when the dataset is linearly separable and the network is a linear network). This issue was mentioned in (Nguyen & Hein, 2017a) and a different loss function was used, but their result only studied the linearly separable case and a subset of the critical points.

In this work, we provide a rather comprehensive study of the loss surface of neural networks for binary classification, by finding a collection of **necessary** and **sufficient** conditions for the loss function to have no bad local minima. On the positive side, we prove that no bad local minima exist

[1]University of Illinois at Urbana-Champaign, [2]Facebook Research. Correspondence to: R. Srikant <rsrikant@illinois.edu>, Ruoyu Sun <ruoyus@illinois.edu>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

under the following conditions: the neurons (i.e. activation functions) are increasing and strictly convex, the neural network is single-layered or is multi-layered with a shortcut-like connection, the surrogate loss function is a smooth version of the hinge loss function, and either the dataset is linearly separable or the positively and negatively labeled samples are located on different subspaces. On the negative side, we provide dozens of counterexamples which show that bad local minima exist when these conditions do not hold. More detailed discussions of the conditions are given as follows.

- For ReLU neurons, we show that the empirical loss has bad local minima. On the positive side, increasing and strictly convex neurons (including smooth versions of ReLUs) can eliminate bad local minima. This is consistent with the practical observation that smooth versions of ReLUs perform better than ReLUs.
- For the loss function, we provide a counterexample in which all local minima of quadratic loss functions have poor training performance in classification tasks. In contrast, the smooth hinge-losses do not have this undesirable property [1].
- For architectures, we have shown that i) pure ReLU feedforward nets have poor training performance; ii) the same conclusion holds even if we use the shortcut connections in ResNet; iii) if the shortcut includes a smooth version of ReLU, then even if the rest of the network uses ReLUs, the landscape of the loss function is much nicer.
- For datasets, we provide necessary conditions without which the network has bad local minima and the question of sufficiency for other neurons is still open.

The outline of this paper is as follows. In Section 2, we present the model and some definitions. In Section 3, we present the main positive results. The necessity of each condition is discussed in Section 4 and proofs are provided in Section 5. Conclusions are presented in Section 6. All other proofs are provided in Appendix.

2. Preliminaries

Network models. Given an input vector x of dimension d, we consider a neural network with L layers for binary classification. We denote by M_l the number of neurons on the l-th layer (note that $M_0 = d$ and $M_L = 1$). We denote the neuron activation function by σ. Let $\boldsymbol{W}_l \in \mathbb{R}^{M_{l-1}\times M_l}$ denote the weight matrix connecting the $(l-1)$-th layer and the l-th layer and $\boldsymbol{b}_l \in \mathbb{R}^{M_l}$ denote the bias vector for the neurons in the l-th layer. Therefore, the output of the network $f : \mathbb{R}^d \to \mathbb{R}$ can be expressed by

$$f(x;\boldsymbol{\theta}) = \boldsymbol{W}_L^\top \sigma\left(...\sigma(\boldsymbol{W}_1^\top x + \boldsymbol{b}_1) + \boldsymbol{b}_{L-1}\right) + \boldsymbol{b}_L,$$

where $\boldsymbol{\theta}$ denotes all parameters in the neural network.

Data distribution. In this paper, we consider binary classification tasks where each sample $(\boldsymbol{X}, Y) \in \mathbb{R}^d \times \{-1, 1\}$ is drawn from an underlying data distribution $\mathbb{P}_{\boldsymbol{X}\times Y}$ defined on $\mathbb{R}^d \times \{-1, 1\}$. The sample $(\boldsymbol{X}, Y)$ is considered positive if $Y = 1$, and negative otherwise. Let $\mathcal{E} = \{\boldsymbol{e}_1, ..., \boldsymbol{e}_d\}$ denote a set of orthonormal basis on the space $\mathbb{R}^d$. Let $\mathcal{U}_+$ and $\mathcal{U}_-$ denote two subsets of $\mathcal{E}$ such that all positive and negative samples are located on the linear span of the set $\mathcal{U}_+$ and $\mathcal{U}_-$, respectively, i.e., $\mathbb{P}_{\boldsymbol{X}|Y}(\boldsymbol{X} \in \text{Span}(\mathcal{U}_+)|Y = 1) = 1$ and $\mathbb{P}_{\boldsymbol{X}|Y}(\boldsymbol{X} \in \text{Span}(\mathcal{U}_-)|Y = -1) = 1$. Let r denote the size of the set $\mathcal{U}_+ \cup \mathcal{U}_-$, r_+ denote the size of the set $\mathcal{U}_+$ and r_- denote the size of the set $\mathcal{U}_-$, respectively.

Loss and error. Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ denote a dataset with n samples, each independently drawn from the distribution $\mathbb{P}_{\boldsymbol{X}\times Y}$. Given a neural network $f(x;\boldsymbol{\theta})$ parameterized by $\boldsymbol{\theta}$ and a loss function $\ell : \mathbb{R} \to \mathbb{R}$, in binary classification tasks[2], we define the **empirical loss** $\hat{L}_n(\boldsymbol{\theta})$ as the average loss of the network f on a sample in the dataset $\mathcal{D}$, i.e.,

$$\hat{L}_n(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^n \ell(-y_i f(x_i;\boldsymbol{\theta})).$$

Furthermore, for a neural network f, we define a binary classifier $g_f : \mathbb{R}^d \to \{-1, 1\}$ of the form $g_f = \text{sgn}(f)$, where the sign function $\text{sgn}(z) = 1$, if $z \geq 0$, and $\text{sgn}(z) = 0$ otherwise. We define the **training error** (also called the **misclassification error**) $\hat{R}_n(\boldsymbol{\theta})$ as the misclassification rate of the neural network $f(x;\boldsymbol{\theta})$ on the dataset $\mathcal{D}$, i.e.,

$$\hat{R}_n(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^n \mathbb{I}\{y_i \neq \text{sgn}(f(x_i;\boldsymbol{\theta}))\},$$

where $\mathbb{I}\{\cdot\}$ is the indicator function. The training error $\hat{R}_n$ measures the classification performance of the network f on the finite samples in the dataset $\mathcal{D}$.

3. Main Results

In this section, we present the main results. We first introduce several important conditions in order to derive the main results, and we will provide further discussions on these conditions in the next section.

3.1. Conditions

To fully specify the problem, we need to specify our assumptions on several components of the model, including:

[1] We do not consider logistic loss since it does not have finite global minima; in contrast, the smooth hinge-loss has finite global minima.

[2] We note that, in regression tasks, the empirical loss is usually defined as $\hat{L}_n(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^n \ell(y_i - f(x_i;\boldsymbol{\theta}))$.

(1) the loss function, (2) the data distribution, (3) the network architecture and (4) the neuron activation function.

Assumption 1 (Loss function) *Let $\ell_p : \mathbb{R} \to \mathbb{R}$ denote a loss function satisfying the following conditions: (1) ℓ_p is a surrogate loss function, i.e., $\ell_p(z) \geq \mathbb{I}\{z \geq 0\}$ for all $z \in \mathbb{R}$, where $\mathbb{I}(\cdot)$ denotes the indicator function; (2) ℓ_p has continuous derivatives up to order p on $\mathbb{R}$; (3) ℓ_p is non-decreasing (i.e., $\ell_p'(z) \geq 0$ for all $z \in \mathbb{R}$) and there exists a positive constant z_0 such that $\ell_p'(z) = 0$ iff $z \leq -z_0$.*

The first condition in Assumption 1 ensures that the training error $\hat{R}_n$ is always upper bounded by the empirical loss $\hat{L}_n$, i.e., $\hat{R}_n \leq \hat{L}_n$. This guarantees that the neural network can correctly classify all samples in the dataset (i.e., $\hat{R}_n = 0$), when the neural network achieves zero empirical loss (i.e., $\hat{L}_n = 0$). The second condition ensures that the empirical loss $\hat{L}_n$ has continuous derivatives with respect to the parameters up to a sufficiently high order. The third condition ensures that the loss function is non-decreasing and $\ell_p'(z) = 0$ is achievable if and only if $z \leq -z_0$. Here, we provide a simple example of the loss function satisfying all conditions in Assumption 1: the polynomial hinge loss, i.e., $\ell_p(z) = [\max\{z + 1, 0\}]^{p+1}$. We note that, in this paper, we use $\hat{L}_n(\boldsymbol{\theta}; p)$ to denote the empirical loss when the loss function is ℓ_p and the network is parametrized by a set of parameters $\boldsymbol{\theta}$. Further results on the impact of loss functions are presented in Section 4.

Assumption 2 (Data distribution) *Assume that for random vectors $\boldsymbol{X}_1, ..., \boldsymbol{X}_{r_+}$ independently drawn from the distribution $\mathbb{P}_{\boldsymbol{X}|Y=1}$ and $\boldsymbol{Z}_1, ..., \boldsymbol{Z}_{r_-}$ independently drawn from the distribution $\mathbb{P}_{\boldsymbol{X}|Y=-1}$, matrices $\left(\boldsymbol{X}_1, ..., \boldsymbol{X}_{r_+}\right) \in \mathbb{R}^{r_+ \times d}$ and $\left(\boldsymbol{Z}_1, ..., \boldsymbol{Z}_{r_-}\right) \in \mathbb{R}^{r_- \times d}$ are full rank matrices with probability one.*

Assumption 2 states that support of the conditional distribution $\mathbb{P}_{\boldsymbol{X}|Y=1}$ is sufficiently rich so that r_+ samples drawn from it will be linearly independent. In other words, by stating this assumption, we are avoiding trivial cases where all the positively labeled points are located in a very small subset of the linear span of $\mathcal{U}_+$. Similarly for the negatively labeled samples.

Assumption 3 (Data distribution) *Assume $|\mathcal{U}_+ \cup \mathcal{U}_-| > \max\{|\mathcal{U}_+|, |\mathcal{U}_-|\}$, i.e., $r > \max\{r_+, r_-\}$.*

Assumption 3 assumes that the positive and negative samples are not located on the same linear subspace. Previous works (Belhumeur et al., 1997; Chennubhotla & Jepson, 2001; Cootes et al., 2001; Belhumeur et al., 1997) have observed that some classes of natural images (e.g., images of faces, handwritten digits, etc) can be reconstructed from lower-dimensional representations. For example, using dimensionality reduction methods such as PCA, one can approximately reconstruct the original image from only a small number of principal components (Belhumeur et al., 1997; Chennubhotla & Jepson, 2001). Here, Assumption 3 states that both the positively and negatively labeled samples have lower-dimensional representations, and they do not exist in the same lower-dimensional subspace. We provide additional analysis in Section 4, showing how our main results generalize to other data distributions.

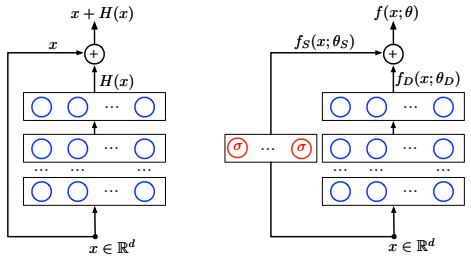
Assumption 4 (Network architecture) *Assume that the neural network f is a single-layered neural network, or more generally, has shortcut-like connections shown in Fig 1 (b), where f_S is a single layer network and f_D is a feedforward network.*

Shortcut connections are widely used in the modern network architectures (e.g., Highway Networks (Srivastava et al., 2015), ResNet (He et al., 2016), DenseNet (Huang et al., 2017), etc.), where the skip connections allow the deep layers to have direct access to the outputs of shallow layers. For instance, in the residual network, each residual block has a identity shortcut connection, shown in Fig 1 (a), where the output of each residual block is the vector sum of its input and the output of a network H.

Instead of using the identity shortcut connection, in this paper, we first pass the input through a single layer network $f_S(x; \boldsymbol{\theta}_S) = a_0 + \boldsymbol{a}^\top \sigma\left(\boldsymbol{W}^\top x\right)$, where vector $\boldsymbol{a}$ denotes the weight vector, matrix $\boldsymbol{W}$ denotes the weight matrix and vector $\boldsymbol{\theta}_S$ denotes the vector containing all parameters in f_S. We next add the output of this network to a network f_D and use the addition as the output of the whole network, i.e., $f(x; \boldsymbol{\theta}) = f_S(x; \boldsymbol{\theta}_S) + f_D(x; \boldsymbol{\theta}_D)$, where vector $\boldsymbol{\theta}_D$ and $\boldsymbol{\theta}$ denote the vector containing all parameters in the network f_D and the whole network f, respectively. We note here that, in this paper, we do not restrict the number of layers and neurons in the network f_D and this means that the network f_D can be a feedforward network introduced in Section 2 or a single layer network or even a constant. In fact, when the network f_D is a single layer network or a constant, the whole network f becomes a single layer network. Furthermore, we note that, in Section 4, we will show that if we remove this connection or replace this shortcut-like connection with the identity shortcut connection, the main result does not hold.

Assumption 5 (Neuron activation) *Assume that neurons $\sigma(z)$ in the network f_S are real analytic and satisfy $\sigma''(z) > 0$ for all $z \in \mathbb{R}$. Assume that neurons in the network f_D are real functions on $\mathbb{R}$.*

In Assumption 5, we assume that neurons in the network f_S are infinitely differentiable and have positive second order derivatives on $\mathbb{R}$, while neurons in the network f_D are real functions. We make the above assumptions to ensure that the loss function $\hat{L}_n(\boldsymbol{\theta}_S, \boldsymbol{\theta}_D; p)$ is partially differentiable w.r.t. the parameters $\boldsymbol{\theta}_S$ in the network f_S up to a sufficiently high order and allow us to use Taylor expansion in the analysis. Here, we list a few neurons



(a) (b)

Figure 1. (a) The identity shortcut connection adopted in the residual network (He et al., 2016). (b) The shortcut-like connection adopted in this paper.

which can be used in the network f_S: softplus neuron, i.e., $\sigma(z) = \log_2(1 + e^z)$, quadratic neuron, i.e, $\sigma(z) = z^2$, etc. We note that neurons in the network f_S and f_D do not need to be of the same type and this means that a more general class of neurons can be used in the network f_D, e.g., threshold neuron, i.e., $\sigma(z) = \mathbb{I}\{z \geq 0\}$, rectified linear unit $\sigma(z) = \max\{z, 0\}$, sigmoid neuron $\sigma(z) = \frac{1}{1+e^{-z}}$, etc. Further discussion on the effects of neurons on the main results are provided in Section 4.

3.2. Main Results

Now we present the following theorem to show that when assumptions 1-5 are satisfied, every local minimum of the empirical loss function has zero training error if the number of neurons in the network f_S are chosen appropriately.

Theorem 1 (Linear subspace data) *Suppose that assumptions 1-5 are satisfied. Assume that samples in the dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n, n \geq 1$ are independently drawn from the distribution $\mathbb{P}_{\boldsymbol{X}\times Y}$. Assume that the number of neurons M in the network f_S satisfies $M \geq 2\max\{\frac{n}{\Delta r}, r_+, r_-\}$, where $\Delta r = r - \max\{r_+, r_-\}$. If $\boldsymbol{\theta}^* = (\boldsymbol{\theta}_S^*, \boldsymbol{\theta}_D^*)$ is a local minimum of the loss function $\hat{L}_n(\boldsymbol{\theta}_S, \boldsymbol{\theta}_D; p)$ and $p \geq 6$, then $\hat{R}_n(\boldsymbol{\theta}_S^*, \boldsymbol{\theta}_D^*) = 0$ holds with probability one.*

Remark: (i) By setting the network f_D to a constant, it directly follows from Theorem 1 that if a single layer network $f_S(x; \boldsymbol{\theta}_S)$ consisting of neurons satisfying Assumption 5 and all other conditions in Theorem 1 are satisfied, then every local minimum of the empirical loss $\hat{L}_n(\boldsymbol{\theta}_S; p)$ has zero training error. (ii) The positiveness of Δr is guaranteed by Assumption 3. In the worst case (e.g., $\Delta r = 1$ and $\Delta r = 2$), the number of neurons needs to be at least greater than the number of samples, i.e., $M \geq n$. However, when the two orthonormal basis sets $\mathcal{U}_+$ and $\mathcal{U}_-$ differ significantly (i.e., $\Delta r \gg 1$), the number of neurons required by Theorem 1 can be significantly smaller than the number of samples (i.e., $n \gg 2n/\Delta r$). In fact, we can show that, when the neuron has quadratic activation function $\sigma(z) = z^2$, the assumption $M \geq 2n/\Delta r$ can be further relaxed such that the number of neurons is independent of the number of samples. We discuss this in the following proposition.

Proposition 1 *Assume that assumptions 1-5 are satisfied. Assume that samples in the dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n, n \geq 1$ are independently drawn from the distribution $\mathbb{P}_{\boldsymbol{X}\times Y}$. Assume that neurons in the network f_S satisfy $\sigma(z) = z^2$ and the number of neurons in the network f_S satisfies $M > r$. If $\boldsymbol{\theta}^* = (\boldsymbol{\theta}_S^*, \boldsymbol{\theta}_D^*)$ is a local minimum of the loss function $\hat{L}_n(\boldsymbol{\theta}_S, \boldsymbol{\theta}_D; p)$ and $p \geq 6$, then $\hat{R}_n(\boldsymbol{\theta}_S^*, \boldsymbol{\theta}_D) = 0$ holds with probability one.*

Remark: Proposition 1 shows that if the number of neuron M is greater than the dimension of the subspace, i.e., $M > r$, then every local minimum of the empirical loss function has zero training error. We note here that although the result is stronger with quadratic neurons, it does not imply that the quadratic neuron has advantages over the other types of neurons (e.g., softplus neuron, etc). This is due to the fact that when the neuron has positive derivatives on $\mathbb{R}$, the result in Theorem 1 holds for the dataset where positive and negative samples are linearly separable. We provide the formal statement of this result in Theorem 2. However, when the neuron has quadratic activation function, the result in Theorem 1 may not hold for linearly separable dataset and we will illustrate this by providing a counterexample in the next section.

As shown in Theorem 1, when the data distribution satisfies Assumption 2 and 3, every local minimum of the empirical loss has zero training error. However, we can easily see that distributions satisfying these two assumptions may not be linearly separable. Therefore, to provide a complementary result to Theorem 1, we consider the case where the data distribution is linearly separable. Before presenting the result, we first present the following assumption on the data distribution.

Assumption 6 (Linear separability) *Assume that there exists a vector $\boldsymbol{w} \in \mathbb{R}^d$ such that $\mathbb{P}_{\boldsymbol{X}\times Y}(Y\boldsymbol{w}^\top X > 0) = 1$.*

In Theorem 2, we will show that when the samples drawn from the data distribution are linearly separable, and the network has a shortcut-like connection shown in Figure 1, all local minima of the empirical loss function have zero training errors if the type of the neuron in the network f_S are chosen appropriately.

Theorem 2 (Linearly separable data) *Suppose that the loss function ℓ_p satisfies Assumption 1 and the network architecture satisfies Assumption 4. Assume that samples in the dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n, n \geq 1$ are independently drawn from a distribution satisfying Assumption 6. Assume that the single layer network f_S has $M \geq 1$ neurons and neurons σ in the network f_S are twice differentiable and*

satisfy $\sigma'(z) > 0$ for all $z \in \mathbb{R}$. If $\boldsymbol{\theta}^* = (\boldsymbol{\theta}_S^*, \boldsymbol{\theta}_D^*)$ is a local minimum of the loss function $\hat{L}_n(\boldsymbol{\theta}_S, \boldsymbol{\theta}_D; p)$, $p \geq 3$, then $\hat{R}_n(\boldsymbol{\theta}_S^*, \boldsymbol{\theta}_D^*) = 0$ holds with probability one.

Remark: Similar to Proposition 1, Theorem 2 does not require the number of neurons to be in scale with the number of samples. In fact, we make a weaker assumption here: the single layer network f_S only needs to have at least one neuron, in contrast to at least r neurons required by Proposition 1. Furthermore, we note here that, in Theorem 2, we assume that neurons in the network f_S have positive derivatives on $\mathbb{R}$. This implies that Theorem 2 may not hold for a subset of neurons considered in Theorem 1 (e.g., quadratic neuron, etc). We will provide further discussions on the effects of neurons in the next section.

4. Discussions

In this section, we discuss the effects of the (1) neuron activation, (2) shortcut-like connections, (3) loss function and (4) data distribution on the main results, respectively. We show that the result may not hold if these assumptions are relaxed.

4.1. Neuron Activations

To begin with, we discuss whether the results in Theorem 1 and 2 still hold if we vary the neuron activation function in the single layer network f_S. Specifically, we consider the following two classes of neurons: (1) softplus class and (2) rectified linear unit (ReLU) class. In the following, for each class of neurons, we show whether the main results hold and provide counterexamples if certain conditions in the main results are violated. We summarize our findings in Table 4.1.

Softplus class contains neurons with real analytic activation functions σ, where $\sigma'(z) > 0, \sigma''(z) > 0$ for all $z \in \mathbb{R}$. A widely used neuron in this class is the softplus neuron, i.e., $\sigma(z) = \log_2(1 + e^z)$, which is a smooth approximation of ReLU. We can see that neurons in this class satisfy assumptions in both Theorem 1 and 2 and this indicates that both theorems hold for the neurons in this class.

ReLU class contains neurons with $\sigma(z) = 0$ for all $z \leq 0$ and $\sigma(z)$ is piece-wise continuous on $\mathbb{R}$. Some commonly adopted neurons in this class include: threshold units, i.e., $\mathbb{I}\{z \geq 0\}$, rectified linear units (ReLU), i.e., $\max\{z, 0\}$ and rectified quadratic units (ReQU), i.e., $[\max\{z, 0\}]^2$. We can see that neurons in this class do not satisfy assumptions in Theorem 1 or 2. In Proposition 2, we show that when the single layer network f_S consists of neurons in the ReLU class, even if all other conditions in Theorem 1 or 2 are satisfied, the loss function can have a bad local minimum.

Proposition 2 *Suppose that assumptions 1 and 4 are satisfied. Assume that neurons in the network f_S satisfy that $\sigma(z) = 0$ for all $z \leq 0$ and $\sigma(z)$ is piece-wise continuous on $\mathbb{R}$. Then there exists a network architecture f_D and a distribution satisfying assumptions in Theorem 1 or 2 such that with probability one, the empirical loss $\hat{L}_n(\boldsymbol{\theta}; p), p \geq 2$ has a local minima $\boldsymbol{\theta}^* = (\boldsymbol{\theta}_S^*, \boldsymbol{\theta}_D^*)$ satisfying $\hat{R}_n(\boldsymbol{\theta}^*) \geq \frac{\min\{n_+, n_-\}}{n}$, where n_+ and n_- are the number of positive and negative samples, respectively.*

Theorem	Soft-plus	ReLU
1	Yes	No
2	Yes	No

Table 1. The result whether Theorem 1 or 2 hold for all neurons in each class. The definition of each class can be found in Section 4.1.

Remark: (i) We note here that the above result holds in the over-parametrized case, where the number of neurons in the network f_S is larger than the number of samples in the dataset. In addition, all counterexamples shown in Section 4.1 hold in the over-parametrized case. (ii) We note here that applying the same analysis, we can generalize the above result to a larger class of neurons satisfying the following condition: there exists a scalar z_1 such that $\sigma(z) =$ constant for all $z \leq z_1$ and $\sigma(z)$ is piece-wise continuous on $\mathbb{R}$. (iii) We note that the training error is strictly non-zero when the dataset has both positive and negative samples and this can happen with probability at least $1 - e^{-\Omega(n)}$.

4.2. Shortcut-like Connections

In this subsection, we discuss whether the main results still hold if we remove the shortcut-like connections or replace them with the identity shortcut connections used in the residual network (He et al., 2016). Specifically, we provide two counterexamples and show that the main results do not hold if the shortcut-like connections are removed or replaced with the identity shortcut connections.

Feed-forward networks. When the shortcut-like connections (i.e., the network f_S in Figure 1(b)) are removed, the network architecture can be viewed as a standard feedforward neural network. We provide a counterexample to show that, for a feedforward network with ReLU neurons, even if the other conditions in Theorem 1 or 2 are satisfied, the empirical loss functions is likely to have a local minimum with non-zero training error. In other words, neither Theorem 1 nor 2 holds when the shortcut-like connections are removed.

Proposition 3 *Suppose that assumption 1 is satisfied. Assume that the feedforward network $f(x; \boldsymbol{\theta})$ has at least one hidden layer and at least one neuron in each hidden layer. If neurons in the network f satisfy that $\sigma(z) = 0$ for all $z \leq 0$ and $\sigma(z)$ is continuous on $\mathbb{R}$, then for any dataset $\mathcal{D}$ with n samples, the empirical loss $\hat{L}_n(\boldsymbol{\theta}; p), p \geq 2$ has*

a local minima $\boldsymbol{\theta}^*$ with $\hat{R}_n(\boldsymbol{\theta}^*) \geq \frac{\min\{n_+, n_-\}}{n}$, where n_+ and n_- are the number of positive and negative samples in the dataset, respectively.

Remark: The result holds for ReLUs, since it is easy to check that the ReLU neuron satisfies the above assumptions.

Identity shortcut connections. As we stated earlier, adding shortcut-like connections to a network can improve the loss surface. However, the shortcut-like connections shown in Fig 1(b) are different from some popular shortcut connections used in the real-world applications, e.g., the identity shortcut connections in the residual network. Thus, a natural question arises: do the main results still hold if we use the identity shortcut connections? To address the question, we provide the following counterexample to show that, when we replace the shortcut-like connections with the identity shortcut connections, even if the other conditions in Theorem 1 are satisfied, the empirical loss function is likely to have a local minimum with non-zero training error. In other words, Theorem 1 does not hold for the identity shortcut connections.

Proposition 4 *Assume that $H : \mathbb{R}^d \to \mathbb{R}^d$ is a feedforward neural network parameterized by $\boldsymbol{\theta}$ and all neurons in H are ReLUs. Define a network $f : \mathbb{R}^d \to \mathbb{R}$ with identity shortcut connections as $f(x; \boldsymbol{a}, \boldsymbol{\theta}, b) = \boldsymbol{a}^\top(x + H(x; \boldsymbol{\theta})) + b$, $\boldsymbol{a} \in \mathbb{R}^d, b \in \mathbb{R}$. Then there exists a distribution $\mathbb{P}_{\boldsymbol{X}\times Y}$ satisfying the assumptions in Theorem 1 such that with probability at least $1 - e^{-\Omega(n)}$, the empirical loss $\hat{L}_n(\boldsymbol{a}, \boldsymbol{\theta}, b; p) = \frac{1}{n}\sum_{i=1}^n \ell(-y_i f(x_i; \boldsymbol{\theta}); p), p \geq 2$ has a local minimum with non-zero training error.*

4.3. Loss Functions

In this subsection, we discuss whether the main results still hold if we change the loss function. We mainly focus on the quadratic loss. We will show that if the loss function is replaced with the quadratic loss, then neither Theorem 1 nor 2 holds.

Quadratic loss. The quadratic loss $\ell(z) = (1 - z)^2$ has been well-studied in prior works. It has been shown that when the loss function is quadratic, under certain assumptions, all local minima of the empirical loss are global minima. However, the global minimum of the quadratic loss does not necessarily have zero misclassification error, even in the realizable case (i.e., the case where there exists a set of parameters such that the network achieves zero misclassification error on the dataset or the data distriubtion). To illustrate this, we provide a simple example where the network is a simplified linear network and the data distribution is linearly separable.

Example 1 *Let the distribution $\mathbb{P}_{\boldsymbol{X}\times Y}$ satisfy that $\mathbb{P}(Y = 1) = \mathbb{P}(Y = -1) = 0.5$, $\mathbb{P}(X = 5/4|Y = 1) = 1$ and $\mathbb{P}_{X|Y=-1}$ is a uniform distribution on the interval $[0, 1]$. For a linear model $f(x; a, b) = ax + b$, $a, b \in \mathbb{R}$, every global minimum (a^*, b^*) of the population loss $L(a, b) = \mathbb{E}_{X\times Y}[(1 - Yf(X; a, b))^2]$ satisfies $\mathbb{P}_{\boldsymbol{X}\times Y}[Y \neq sgn(f(X; a^*, b^*))] \geq 1/16$.*

Remark: The proof of the above result in Appendix B.4 is very straightforward. We have only provided it there since we are unable to find a reference which explicitly states such a result, but we will not be surprised if this result has been known to others. This example shows that every global minimum of the quadratic loss has non-zero misclassification error, although the linear model is able to achieve zero misclassification error on this data distribution. Similarly, one can easily find datasets under which all global minima of the quadratic loss have non-zero training error.

In addition, we provide two examples in Appendix B.5 and show that, when the loss function is replaced with the quadratic loss, even if the other conditions in Theorem 1 or 2 are satisfied, every global minimum of the empirical loss has a training error larger than $1/8$ with a positive probability. In other words, our main results do hold for the quadratic loss.

The following observation may be of independent interest. Different from the quadratic loss, the loss functions conditioned in Assumption 1 have the following two properties: (i) the minimum empirical loss is zero if and only if there exists a set of parameters achieving zero training error; (ii) every global minimum of the empirical loss has zero training error in the realizable case.

Proposition 5 *Let $f : \mathbb{R}^d \to \mathbb{R}$ denote a feedforward network parameterized by $\boldsymbol{\theta}$ and let the dataset have n samples. When the loss function ℓ_p satisfies Assumption 1 and $p \geq 1$, we have $\min_{\boldsymbol{\theta}} \hat{L}_n(\boldsymbol{\theta}; p) = 0$ if and only if $\min_{\boldsymbol{\theta}} \hat{R}_n(\boldsymbol{\theta}) = 0$. Furthermore, if $\min_{\boldsymbol{\theta}} \hat{R}_n(\boldsymbol{\theta}) = 0$, every global minimum $\boldsymbol{\theta}^*$ of the empirical loss $\hat{L}_n(\boldsymbol{\theta}; p)$ has zero training error, i.e., $\hat{R}_n(\boldsymbol{\theta}^*) = 0$.*

Remark: We note that the network does not need to be a feedforward network. In fact, the same results hold for a large class of network architectures, including both architectures shown in Fig 1. We provide additional analysis in Appendix B.6.

4.4. Open Problem: Datasets

In this paper, we have mainly considered a class of non-linearly separable distribution where positive and negative samples are located on different subspaces. We show that if the samples are drawn from such a distribution, under certain additional conditions, all local minima of the empirical loss have zero training errors. However, one may ask: how well does the result generalize to other non-linearly separable distributions or datasets? Here, we partially answer this

question by presenting the following necessary condition on the dataset so that Theorem 1 can hold.

Proposition 6 *Suppose that assumptions 1, 4 and 5 are satisfied. For any feedforward architecture $f_D(x;\boldsymbol{\theta}_D)$, every local minimum $\boldsymbol{\theta}^* = (\boldsymbol{\theta}_S^*, \boldsymbol{\theta}_D^*)$ of the empirical loss function $\hat{L}_n(\boldsymbol{\theta}_S, \boldsymbol{\theta}_D; p)$, $p \geq 6$ satisfies $\hat{R}_n(\boldsymbol{\theta}^*) = 0$ **only if** the matrix $\sum_{i=1}^n \lambda_i y_i x_i x_i^\top$ is neither positive nor negative definite for all sequences $\{\lambda_i \geq 0\}_{i=1}^n$ satisfying $\sum_{i:y_i=1} \lambda_i = \sum_{i:y_i=-1} \lambda_i > 0$ and $\| \sum_{i=1}^n \lambda_i y_i x_i \|_2 = 0$.*

Remark: The proposition implies that when the dataset does not meet this necessary condition, there exists a feedforward architecture f_D such that the empirical loss function has a local minimum with a non-zero training error. Therefore, Theorem 1 no longer holds when Assumption 2 or 3 is removed.

5. Proofs

In this section, we provide the proof of Theorem 2. Before presenting the proof of Theorem 2, we first present an important lemma. This lemma present a necessary condition such that every local minimum of the empirical loss function has to satisfy.

Lemma 1 *Assume that neurons σ in the network f_S are twice differentiable and the loss function $\ell : \mathbb{R} \to \mathbb{R}$ has a continuous derivative on $\mathbb{R}$ up to the third order. If $n \geq 1$ and parameters $\boldsymbol{\theta}^* = (\boldsymbol{\theta}_S^*, \boldsymbol{\theta}_D^*)$ denote a local minimum of the loss function $\hat{L}_n(\boldsymbol{\theta})$, then for any $j = 1, ..., M$,*

$$\sum_{i=1}^n \ell'(-y_i f(x_i;\boldsymbol{\theta}^*)) y_i \sigma'(\boldsymbol{w}_j^{*\top} x_i) x_i = \mathbf{0}_d.$$

Proof: We first recall some notations defined in the paper. The output of the neural network is

$$f(x;\boldsymbol{\theta}) = f_S(x;\boldsymbol{\theta}_S) + f_D(x;\boldsymbol{\theta}_D),$$

where $f_S(x;\boldsymbol{\theta}_S)$ is the single layer neural network parameterized by $\boldsymbol{\theta}_S$, i.e.,

$$f_S(x;\boldsymbol{\theta}_S) = a_0 + \sum_{j=1}^M a_j \sigma\left(\boldsymbol{w}_j^\top x\right),$$

and $f_D(x;\boldsymbol{\theta}_D)$ is a deep neural network parameterized by $\boldsymbol{\theta}_D$. The empirical loss function is given by

$$\hat{L}_n(\boldsymbol{\theta}) = \hat{L}_n(\boldsymbol{\theta}_S, \boldsymbol{\theta}_D) = \frac{1}{n}\sum_{i=1}^n \ell(-y_i f(x_i;\boldsymbol{\theta})).$$

Since the loss function ℓ has a continuous derivative on $\mathbb{R}$ up to the third order, neurons σ in the network f_S are twice differentiable, then the gradient vector $\nabla_{\boldsymbol{\theta}_S} \hat{L}_n(\boldsymbol{\theta}_S^*, \boldsymbol{\theta}_D^*)$ and the Hessian matrix $\nabla^2_{\boldsymbol{\theta}_S} \hat{L}_n(\boldsymbol{\theta}_S^*, \boldsymbol{\theta}_D^*)$ exists. Furthermore, by the assumption that $\boldsymbol{\theta}^* = (\boldsymbol{\theta}_S^*, \boldsymbol{\theta}_D^*)$ is a local minima of the loss function $\hat{L}_n(\boldsymbol{\theta})$, then we should have for $j = 1, ..., M$,

$$\begin{aligned}\mathbf{0}_d &= \nabla_{\boldsymbol{w}_j} L_n(\boldsymbol{\theta}^*) \\ &= -a_j^* \sum_{i=1}^n \ell'(-y_i f(x_i;\boldsymbol{\theta}^*)) y_i \sigma'(\boldsymbol{w}_j^{*\top} x_i) x_i. \qquad (1)\end{aligned}$$

Now we need to prove that if $\boldsymbol{\theta}^*$ is a local minima, then for $\forall j \in [M]$, we should obtain

$$\left\| \sum_{i=1}^n \ell'(-y_i f(x_i;\boldsymbol{\theta}^*)) y_i \sigma'(\boldsymbol{w}_j^{*\top} x_i) x_i \right\|_2 = 0.$$

We prove it by contradiction. Assume that there exists $j \in [M]$ such that

$$\left\| \sum_{i=1}^n \ell'(-y_i f(x_i;\boldsymbol{\theta}^*)) y_i \sigma'(\boldsymbol{w}_j^{*\top} x_i) x_i \right\|_2 \neq 0.$$

Then by equation (1), we have $a_j^* = 0$. Now, we consider the following Hessian matrix $H(a_j, \boldsymbol{w}_j)$. Since $\boldsymbol{\theta}^*$ is a local minima of the loss function $\hat{L}_n(\boldsymbol{\theta})$, then the matrix $H(a_j, \boldsymbol{w}_j)$ should be positive semidefinite at $(a_j^*, \boldsymbol{w}_j^*)$. By $a_j^* = 0$, we have

$$\begin{aligned}&\nabla^2_{\boldsymbol{w}_j} L_n(\boldsymbol{\theta}^*) = \mathbf{0}_{d\times d}, \\ &\frac{\partial \left[\nabla_{w_j} L_n(\boldsymbol{\theta}^*)\right]}{\partial a_j} = -\sum_{i=1}^n \ell'(-y_i f(x_i;\boldsymbol{\theta}^*)) y_i \sigma'(\boldsymbol{w}_j^{*\top} x_i) x_i.\end{aligned}$$

In addition, we have

$$\begin{aligned}\frac{\partial^2 L_n(\boldsymbol{\theta}^*)}{\partial a_j^2} &= \frac{\partial}{\partial a_j}\left[\sum_{i=1}^n \ell'(-y_i f(x_i;\boldsymbol{\theta}^*))(-y_i \sigma(\boldsymbol{w}_j^{*\top} x_i))\right] \\ &= \sum_{i=1}^n \ell''(-y_i f(x_i;\boldsymbol{\theta}^*)) \sigma^2(\boldsymbol{w}_j^{*\top} x_i).\end{aligned}$$

Since the matrix $H(a_j^*, \boldsymbol{w}_j^*)$ is positive semidefinite, then for any $\alpha \in \mathbb{R}$ and $\boldsymbol{\omega} \in \mathbb{R}^d$,

$$\begin{pmatrix}\alpha & \boldsymbol{\omega}^\top\end{pmatrix} H(a_j^*, \boldsymbol{w}_j^*) \begin{pmatrix}\alpha \\ \boldsymbol{\omega}\end{pmatrix} \geq 0.$$

Thus, by setting

$$\boldsymbol{\omega} = \sum_{i=1}^n \ell'(-y_i f(x_i;\boldsymbol{\theta}^*)) y_i \sigma'(\boldsymbol{w}_j^{*\top} x_i) x_i,$$

then

$$\begin{aligned}&\begin{pmatrix}\alpha & \omega^\top\end{pmatrix} H(a_j^*, \boldsymbol{w}_j^*) \begin{pmatrix}\alpha \\ \omega\end{pmatrix} \\ &= \alpha^2 \sum_{i=1}^n \ell''(-y_i f(x_i;\boldsymbol{\theta}^*))\sigma^2({\boldsymbol{w}_j^*}^\top x_i) \\ &\quad - \alpha \left\| \sum_{i=1}^n \ell'(-y_i f(x_i;\boldsymbol{\theta}^*)) y_i \sigma'({\boldsymbol{w}_j^*}^\top x_i) x_i \right\|_2^2.\end{aligned}$$

Furthermore, since we assume that

$$\left\| \sum_{i=1}^n \ell'(-y_i f(x_i;\boldsymbol{\theta}^*)) y_i \sigma'({\boldsymbol{w}_j^*}^\top x_i) x_i \right\|_2^2 > 0,$$

then clearly, there exists α such that

$$\begin{pmatrix}\alpha & \boldsymbol{\omega}^\top\end{pmatrix} H(a_j^*, \boldsymbol{w}_j^*) \begin{pmatrix}\alpha \\ \boldsymbol{\omega}\end{pmatrix} < 0,$$

and this leads to the contradiction. □

Now we present the proof of Theorem 2.

Proof: The empirical loss function is given by

$$\hat{L}_n(\boldsymbol{\theta}; p) = \hat{L}_n(\boldsymbol{\theta}_S, \boldsymbol{\theta}_D; p) = \frac{1}{n}\sum_{i=1}^n \ell_p(-y_i f(x_i;\boldsymbol{\theta})).$$

By the assumption that $\boldsymbol{\theta}^* = (\boldsymbol{\theta}_S^*, \boldsymbol{\theta}_D^*)$ is a local minima and by the necessary condition presented in Lemma 1, we have

$$\sum_{i=1}^n \ell_p'(-y_i f(x_i;\boldsymbol{\theta}^*)) y_i \sigma'({\boldsymbol{w}_j^*}^\top x_i) x_i = \mathbf{0}_d.$$

Thus, for any $\boldsymbol{w} \in \mathbb{R}^d$ and any $j \in [M]$, we have

$$\sum_{i=1}^n \ell_p'(-y_i f(x_i;\boldsymbol{\theta}^*)) \sigma'({\boldsymbol{w}_j^*}^\top x_i) y_i (\boldsymbol{w}^\top x_i) = 0.$$

Furthermore, by assumption

$$\ell_p'(z) \geq 0$$

and the equality holds if and only if $z \leq -z_0$. Thus, by assumption that $\sigma'(z) > 0$ for all $z \in \mathbb{R}$ and assumption that there exists a vector $\mathbb{P}_{\boldsymbol{X}\times Y}(Y\boldsymbol{w}^\top X > 0) = 1$, then there exists and positive constant $c > 0$ such that

$$y_i(\boldsymbol{w}^\top x_i) > c > 0, \quad \forall i \in [n].$$

Thus, we have

$$\begin{aligned}0 &= \sum_{i=1}^n \ell_p'(-y_i f(x_i;\boldsymbol{\theta}^*)) \sigma'({\boldsymbol{w}_j^*}^\top x_i) y_i (\boldsymbol{w}^\top x_i) \\ &\geq c \sum_{i=1}^n \ell_p'(-y_i f(x_i;\boldsymbol{\theta}^*)) \sigma'({\boldsymbol{w}_j^*}^\top x_i) \\ &\geq 0,\end{aligned}$$

where the equality holds if and only if $\ell_p'(-y_i f(x_i;\boldsymbol{\theta}^*)) = 0$ for all $i \in [n]$. Equivalently, if $\boldsymbol{\theta}^*$ is a local minima, then $y_i f(x_i;\boldsymbol{\theta}^*) \geq z_0 > 0$ for all $i \in [n]$. This indicates that $L_n(\boldsymbol{\theta}^*; p) = \hat{R}_n(\boldsymbol{\theta}^*) = 0$. □

6. Conclusions

In this paper, we studied the loss surface of a smooth version of the hinge loss function in binary classification problems. We provided conditions under which the neural network has zero misclassification error at all local minima and also provide counterexamples to show that when some of these assumptions are relaxed, the result may not hold. Further work involves exploiting our results to design efficient training algorithms classification tasks using neural networks.

7. Acknowledgement

Research supported by the following grants: NSF NeTS 1718203, NSF CPS ECCS 1739189, a start-up grant from University of Illinois Urbana-Champaign.

References

Andoni, A., Panigrahy, R., Valiant, G., and Zhang, L. Learning polynomials with neural networks. In *ICML*, 2014.

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

Belhumeur, P. N., Hespanha, J. P., and Kriegman, D. J. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7):711–720, 1997.

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

Bottou, L. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pp. 177–186. Springer, 2010.

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

Chennubhotla, C. and Jepson, A. Sparse pca. extracting multi-scale structure from data. In *ICCV*, volume 1, pp. 641–647. IEEE, 2001.

Choromanska, A., Henaff, M., Mathieu, M., Arous, G., and LeCun, Y. The loss surfaces of multilayer networks. In *AISTATS*, 2015.

Cootes, T. F., Edwards, G. J., and Taylor, C. J. Active appearance models. *IEEE Transactions on pattern analysis and machine intelligence*, 23(6):681–685, 2001.

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., and Tian, Y. When is a convolutional filter easy to learn? *arXiv preprint arXiv:1709.06129*, 2017.

Freeman, C. D. and Bruna, J. Topology and geometry of half-rectified network optimization. *ICLR*, 2016.

Gautier, A., Nguyen, Q. N., and Hein, M. Globally optimal training of generalized polynomial neural networks with nonlinear spectral methods. In *Advances in Neural Information Processing Systems*, pp. 1687–1695, 2016.

Goel, S. and Klivans, A. Learning depth-three neural networks in polynomial time. *arXiv preprint arXiv:1709.06010*, 2017.

Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A., and Bengio, Y. Maxout networks. *arXiv preprint arXiv:1302.4389*, 2013.

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

Hardt, M. and Ma, T. Identity matters in deep learning. *ICLR*, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016.

Huang, G., L., Z., W., K. Q., and M., L. V. D. Densely connected convolutional networks. In *CVPR*, 2017.

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

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

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

LeCun, Y., Bengio, Y., and Hinton, G. E. Deep learning. *Nature*, 521(7553):436, 2015.

Li, Y. and Yuan, Y. Convergence analysis of two-layer neural networks with relu activation. In *NIPS*, pp. 597–607, 2017.

Li, Y., Yosinski, J., Clune, J., Lipson, H., and Hopcroft, J. Convergent learning: Do different neural networks learn the same representations? *arXiv preprint arXiv:1511.07543*, 2015.

Nguyen, Q. and Hein, M. The loss surface and expressivity of deep convolutional neural networks. *arXiv preprint arXiv:1710.10928*, 2017a.

Nguyen, Q. and Hein, M. The loss surface of deep and wide neural networks. *arXiv preprint arXiv:1704.08045*, 2017b.

Sedghi, H. and Anandkumar, A. Provable methods for training neural networks with sparse connectivity. *arXiv preprint arXiv:1412.2693*, 2014.

Soltanolkotabi, M. Learning relus via gradient descent. In *NIPS*, pp. 2004–2014, 2017.

Soltanolkotabi, M., Javanmard, A., and Lee, J. D. Theoretical insights into the optimization landscape of over-parameterized shallow neural networks. *arXiv preprint arXiv:1707.04926*, 2017.

Soudry, D. and Hoffer, E. Exponentially vanishing sub-optimal local minima in multilayer neural networks. *arXiv preprint arXiv:1702.05777*, 2017.

Srivastava, R. K., Greff, K., and Schmidhuber, J. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

Wan, L., Zeiler, M., Zhang, S., Le Cun, Y., and Fergus, R. Regularization of neural networks using dropconnect. In *ICML*, pp. 1058–1066, 2013.

Yun, C., Sra, S., and Jadbabaie, A. Global optimality conditions for deep neural networks. *arXiv preprint arXiv:1707.02444*, 2017.

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.