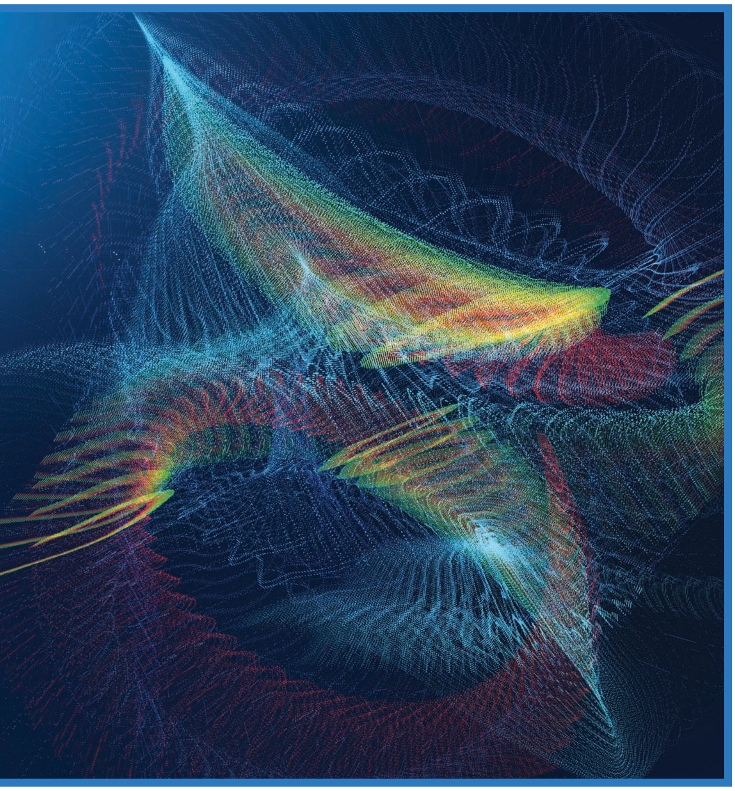
Ruoyu Sun, Dawei Li, Shiyu Liang, Tian Ding,
and Rayadurgam Srikant

The Global Landscape of Neural Networks

An overview



©ISTOCKPHOTO.COM/IN-FUTURE

One of the major concerns for neural network training is that the nonconvexity of the associated loss functions may cause a bad landscape. The recent success of neural networks suggests that their loss landscape is not too bad, but what specific results do we know about the landscape? In this article, we review recent findings and results on the global landscape of neural networks.

First, we point out that wide neural nets may have suboptimal local minima under certain assumptions. Second, we discuss a few rigorous results on the geometric properties of wide networks, such as "no bad basin," and some modifications that eliminate suboptimal local minima and/or decreasing paths to infinity. Third, we discuss the visualization and empirical explorations of the landscape for practical neural nets. Finally, we briefly discuss some convergence results and their relation to landscape results.

Introduction

Deep neural networks have led to remarkable empirical successes in various artificial intelligence tasks, sparking an interest in the theory behind their architectures and training. In the early days, when the power of neural networks was not fully harnessed, researchers favored models, such as support vector machines, that could be studied using convex optimization techniques. A major concern in the case of neural networks is that the nonconvexity of the associated loss functions may cause complicated and strange optimization landscapes. However, recent experience shows that neural networks can often be trained to find the global minima of appropriately chosen loss functions; thus, it is of great interest to understand the loss landscape of neural networks.

A closely related problem is to understand the landscape of the objectives in nonconvex matrix problems. In this context, it has been established that the landscape is benign (e.g., every local minimum is a global minimum) for quite a few matrix problems, such as matrix completion and phase retrieval, under certain assumptions (see, e.g., [9] for a survey). Although there are still many cases that remain difficult to analyze, there is

Digital Object Identifier 10.1109/MSP.2020.3004124
Date of current version: 2 September 2020

much optimism that nonconvex matrix problems (under proper assumptions) often have benign landscapes.

For the landscape of neural networks, the status is less clear. One might be interested in getting a "yes" or "no" answer to questions, such as "Does a neural network have suboptimal local minima?" or "Can a neural net problem be solved to find global minima?" From the many positive headlines, one may get the impression that these questions have positive answers. However, one should distinguish compressed claims and rigorous theoretical results, and it requires some effort to understand the precise results. We hope this article can explain the existing results in a coherent way so that they are relatively easy to understand. Throughout the article, we use "local-min" as an abbreviation of "local minimum" and "global-min" as an abbreviation of "global minimum."

Summary

The goal of this survey is to provide an overview of the recent progress on the global landscape of neural networks. The central questions to answer are as follows.

- *Question 1*: How can the good performance of neural net optimization algorithms be explained?
- *Question 2 (rigorous evidence for Question 1)*: To prove a rigorous result, what conditions are needed, and what property can be established?
- *Question 3*: How can the system be designed so that a rigorous theory can be established?

Based on these high-level questions, we organize the existing results in a flow as follows.

- An initial explanation is Hypothesis 1: "Every local-min is a global-min." The first rigorous evidence is that deep linear networks have no suboptimal local minima under mild conditions (see the "Linear Neural Networks" section).
- However, practitioners find that narrow neural nets cannot be solved well, while overparameterized neural nets can. Thus, researchers believe that a crucial condition is "overparameterization." Results on linear networks do not utilize this condition and, thus, are not enough to explain practice.

The theory of machine learning (for supervised learning) consists of three parts: representation, optimization, and generalization.

- For nonlinear, overparameterized neural nets, a suboptimal local-min can exist under certain assumptions (see the "Does Overparameterization Eliminate Bad Local Minima for Smooth Neurons?" section). An alternative Hypothesis 2 is that local descent algorithms "avoid" suboptimal local minima in neural net training.
- One explanation of Hypothesis 2 is the following. For many overparameterized neural nets, no "bad valleys/basins" exist (see the "The Absence of Bad Valleys and Basins" section). Thus, even if suboptimal local minima exist, they cannot be strong attractors, and, thus, iterates will not be attracted to them. (This part lacks formal results; see the "Escaping Saddle Points" section).
- With stronger assumptions (e.g., ultrawide nets), gradient descent (GD) can avoid reaching the area with suboptimal local minima, thus converging to global minima (see the "Algorithmic Analysis for Ultrawide Networks" section). Due to space constraints, we only touch the surface of this subarea in this article.
- The research described previously assumes no modification of the neural net landscape. If we are allowed to design the landscape (e.g., by adding regularizers), then proving that "every local-min is global-min" becomes possible for a wide range of neural networks (see the "Making Modifications to Eliminate Bad Local Minima" section). We further discuss a result that ensure the absence of both bad local minima and decreasing paths to infinity (see the "Eliminating Bad Local Minima and Decreasing the Paths to Infinity" section).
- Finally, what is the lesson for empirical training? The successful training of neural nets requires proper initialization, batch normalization, residual connection, and wide/deep networks (see [41] for a more thorough survey). The current article focuses on one lesson: large width is important for successful training. For practitioners, many theoretical results in this article can be viewed as evidence of this lesson. For theoreticians, the reviewed results provide a more precise understanding of the benefit of width.

The big picture: The role of landscape analysis

Landscape analysis has been a subject of study since 1980s; see [7] for an overview. The concern that GD can get stuck at bad local minima has been around for a long time. For instance, Minsky and Papert commented in *Perceptrons* (expanded edition) [47] that "they speak as though becoming trapped on local maxima were rarely a serious problem."

Bianchi and Gori [7] argued that, to address Minsky and Papert's comment, it is "very interesting to investigate the presence of local minima." Our survey can be viewed as a modern version of the survey [7] that includes new results and new understanding, especially the results on deep networks. In particular, we point out that a main claim reviewed in [7] that "overparameterized one-hidden-layer networks have no suboptimal local minima" is not rigorous.

The theory of machine learning (for supervised learning) consists of three parts: representation, optimization, and generalization. One way to interpret this partition is via the lens of error decomposition: the test error can be decomposed as the sum of representation error, optimization error, and generalization error. Landscape analysis is an important component of understanding the optimization error. The optimization error refers to $F(\hat{w}) - F^*$, where F is the loss function, $\hat{w}$ is the solution (i.e., the parameters of the neural network) found by an algorithm, and F^* is the globally minimal loss. Define w^∞ to be a converged solution if the algorithm runs for infinite time and assume it converges. The optimization error can be further decomposed into two parts

$$[F(\hat{w}) - F(w^{\infty})] + [F(w^{\infty}) - F^{*}].$$

The first part is the "nonconvergence error," which occurs because either the algorithm is intrinsically not convergent or the algorithm has not converged yet due to limited running time. It is often reasonable to assume w^{∞} is a stationary point or even a local-min. The second part is the "infinite-time error," which indicates how far away the converged value is from the global-min value. If every local-min is a global-min, and w^{∞} is a local-min, then this term becomes zero. Proving the convergence of an algorithm is a central task of classical optimization, but it often does not cover the "infinite-time error"; in contrast, in landscape analysis, we assume the algorithm can converge and focus on the "infinite-time error."

Therefore, landscape analysis provides an understanding of the fundamental limit of the loss function, and it is somewhat similar to Shannon's capacity bound: it indicates how well an algorithm can possibly perform with a long training time. Although our focus is on landscape analysis, we briefly discuss how a good landscape can possibly lead to convergence to the global-min. Another important topic is the relation of optimization and generalization, such as implicit regularization (e.g., [48] and [49]) and the conjecture that wide minima generalize better (e.g., [50]). Due to space constraints, we do not discuss generalization in this article.

Models

In this section, we present the optimization formulation for a supervised learning problem. Consider input instances $x_i \in \mathbb{R}^{d_x}$ and output instances $y_i \in \mathbb{R}^{d_y}, i = 1, \ldots, n$, where n is the number of samples. The goal is to build a model that can predict y_i based on x_i. We use a neural network $f_\theta : \mathbb{R}^{d_x} \to \mathbb{R}^{d_y}$ to produce a prediction $\hat{y}$ based on an input x. For most parts of the article, we consider a fully connected neural network,

$$\begin{aligned} f_\theta(x) = W_L \phi(W_{L-1} \ldots \\ \phi(W_2 \phi(W_1 x + b_1) + b_2) \cdots + b_{L-1}), \end{aligned} \tag{1}$$

where $\phi : \mathbb{R} \to \mathbb{R}$ is the neuron activation function (or simply "activation"); W_j is a matrix of dimension $d_j \times d_{j-1}$ and $j = 1, \ldots, L$; and $\theta = (W_1, b_1, \ldots, W_{L-1}, b_{L-1}, W_L)$ is the collection of all parameters. Note that we denote $d_0 = d_x$ and $d_L = d_y$. We will use $\phi(Z)$ to denote a matrix with each entry $\phi(Z)_{ij}$ being $\phi(Z_{ij})$.

For a certain loss $\ell(\cdot,\cdot)$, the problem of finding the optimal parameters can be written as

$$\min_{\theta} F(\theta) \triangleq \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f_\theta(x_i)). \tag{2}$$

For regression problems, $\ell(y, z)$ is often the quadratic loss $\ell(y, z) = \| y - z \|^2$. For binary classification problems, a popular choice of ℓ is the logistic loss $\ell(y, z) = \log(1 + \exp(-yz))$.

Finally, we present a few standard definitions. We say $\bar{\theta}$ is a global minimum (or global-min for short) of function F if and only if $F(\bar{\theta}) \le F(\theta), \forall \theta$. We say $\bar{\theta}$ is a critical point of function F if and only if $\nabla F(\bar{\theta}) = 0$. We say $\hat{\theta}$ is a local minimum (or local-min for short) of a function $F(\theta)$ if and only if there exists an open set B that contains $\hat{\theta}$ such that $F(\hat{\theta}) \le F(\theta), \forall \theta \in B$. We say $\hat{\theta}$ is a strict local-min if the inequality is strict for any other $\theta \in B$. The local maximum can be defined in a similar way (replacing $\le$ by $\ge$). We say $\hat{\theta}$ is a saddle point if and only if it is a critical point and neither a local-min nor a local maximum.

Linear neural networks

The initial hypothesis is that "every local-min is global-min" in practical neural nets. The results on linear neural networks were considered early evidence (though not strong) and, thus, historically important. Besides the historical reasons, studying linear networks can help develop technical tools. We remark that linear neural networks are rarely used in practice since their representation power is the same as a linear model, so nontheory readers can skip this section if not interested.

A toy example

We consider the simplest linear neural network problem

$$\min_{v, w \in \mathbb{R}} (vw - 1)^2.$$

This is a nonconvex problem, but it is easy to prove that every local-min is a global-min. We plot the function and its contour in Figure 1.

Hamiltonian of a spin-glass system

Choromanska et al. [11] analyzed the global landscape of multilayer networks. The motivation was to study a multilayer network with rectified linear unit (ReLU) activations, but the ReLU activations are removed by adding a somewhat unrealistic assumption, thus essentially converting the network into a multilayer linear network. Under a few other assumptions, the loss function is transformed to the polynomial function $\Sigma_{i_1, \ldots, i_L = 1}^{p} X_{i_1, i_2, \ldots, i_L} w_{i_1} \cdots w_{i_L}$ with Gaussian random coefficients $X_{i_1, i_2, \ldots, i_L}$.

Definition 1

Index: *The index of a critical point is the number of negative eigenvalues of the Hessian at this point.*

Choromanska et al. [11] computed the limit of the expected number of stationary points with a given index as the width p goes to infinity. Based on the calculations, they described a layered structure for stationary points with different indices: low-index stationary points (including local minima) are closer to global minima than high-index stationary points. (The precise statement is highly technical and omitted here.) While their neural network model is somewhat far from practice, the description of the landscape is rather unique and not seen in other works. We remark that there might be a tradeoff between intuition and rigor: [11] covers not only local minima but also other critical points and, thus, contains "more intuition" than works that only study local minima; meanwhile, the result requires more unrealistic assumptions than other works as well. A reader may find this result more

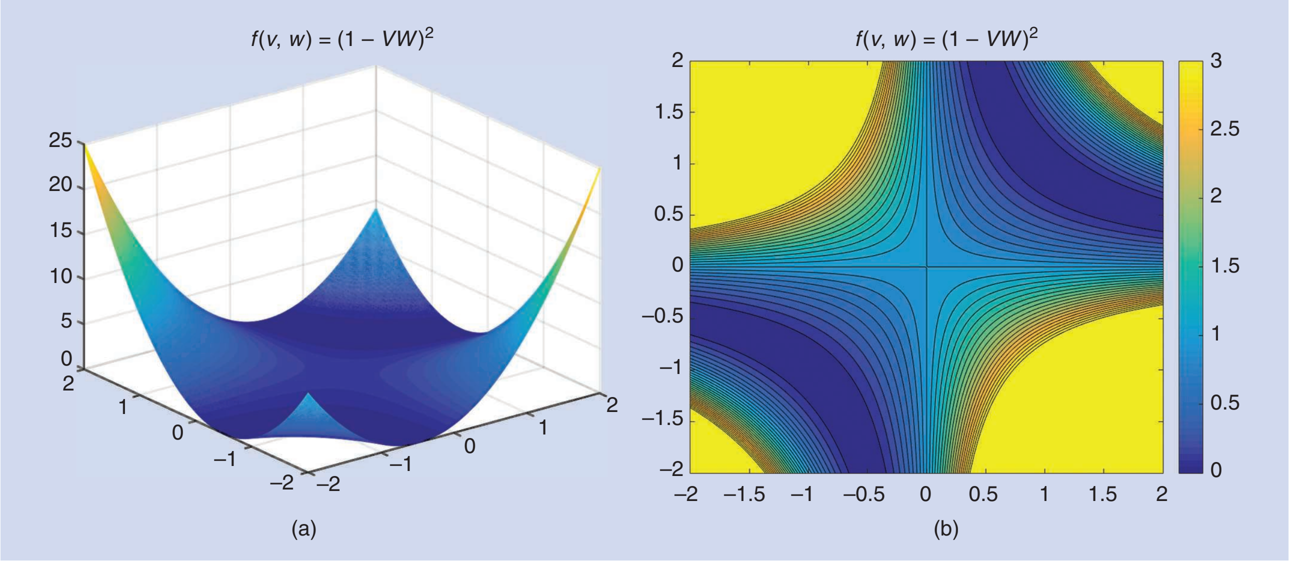


FIGURE 1. The (a) 3D image and (b) 2D contours of the loss surface of $f(v, w) = (1 - vw)^2$. All global minima lie in the two curved regions in dark blue.

interesting or less interesting, depending on how much rigor he or she expects.

Deep linear networks

Kawaguchi [24] extended an early work [6] on two-layer linear networks to deep linear networks, showing that every local-min is a global-min. More specifically, the following problem was studied:

$$\mathrm{P}_1: \quad \min_{W_1, W_2, \ldots, W_L} \| Y - W_L W_{L-1} \ldots W_1 X \|_F^2, \tag{3}$$

where $W_i \in \mathbb{R}^{d_i \times d_{i-1}}, i = 1, \ldots, L$. Reference [24 Th. 2.3] is the first landscape result on this problem; in Theorem 1, we state the slightly stronger version from [35].

Theorem 1

Suppose X and Y have full-row rank; then, every local minimum of (3) is a global minimum.

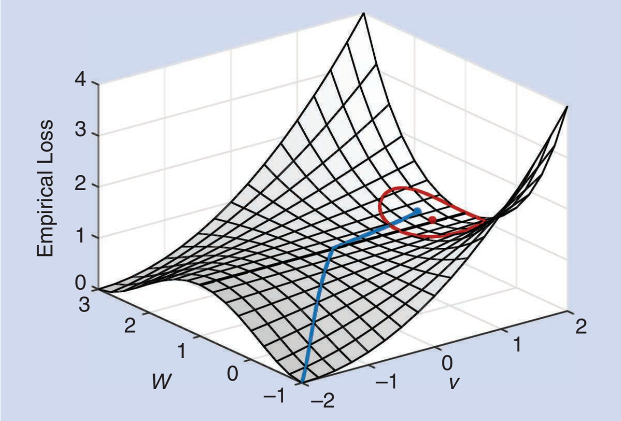


FIGURE 2. The loss surface of $\min_{v,w \in \mathbb{R}} (y - v\phi(wx))^2$, with $(x, y) = (1, 1)$ and $\phi(z) = -(z-1)^2/8$. The red point $(v, W) = (1, 1)$ is a suboptimal local-min, but it is not a strict local-min (flat along one direction and curved along the orthogonal direction). (Source: reprinted from [29].)

The proof of [24] is rather complicated, and [35] provides a simpler and more intuitive proof. The idea of [35] is to view the optimization problem (3) as a reparameterization of a "mother" problem $\mathrm{P}_2 : \min_{\mathrm{rank}(R) \le p} \| Y - RX \|_F^2$, where $p = \min\{d_0, d_1, \ldots, d_L\}$. Note that the effective search space of the original problem $\{W_L \ldots W_2 W_1 \mid W_i \in \mathbb{R}^{d_i \times d_{i-1}}\}$ is the same as the new search space $\{R : \mathrm{rank}(R) \le p\}$, which is why we call P_2 a *mother problem*. The first step is to prove that any local-min of P_1 achieves the value of a local-min of P_2, and the second step is to prove that P_2 has no suboptimal local-min.

References [44] and [45] provide a more precise characterization of the critical points of deep linear networks. Due to space constraints, we do not review the details here.

Overparameterized networks

A major goal of theoretical research is to identify the critical factors of modern neural networks that contribute to successful training. Currently, it is commonly believed that large width is one such factor. One piece of evidence is the empirical observation that wide networks are easier to train than narrow networks (e.g., achieving smaller training and test errors). Another piece of evidence is that pruned models can achieve similar performance to the original model (e.g., [21]), implying that there are many redundant parameters to help optimization. It is, thus, an interesting theoretical question whether overparameterization indeed leads to a benign landscape. In this section, we discuss the landscape of deep overparameterized networks (more precisely, wide networks).

A toy example: A single neuron

As a toy example, we consider the case $n = d_1 = 1$, i.e., a single sample and a nonlinear network with a single neuron. Suppose the associated objective function is $\min_{v,w \in \mathbb{R}} (1 - v\phi(w))^2$. We visualize the landscape for a special activation in Figure 2, and it shows that there are infinitely many suboptimal local minima. The following result describes the landscape of this toy model for general activation functions.

Proposition 1

Suppose $x, y \in \mathbb{R}\backslash\{0\}$. *The function* $F(w,v) = (y - v\phi(wx))^2$, *where* $v, w \in \mathbb{R}$ *has no suboptimal local minima if and only if the following condition holds: if* $\phi(t) = 0$, *then t is not a local-min or local maximum of* ϕ *[13].*

This result shows that the landscape depends on the neuron activation. For instance, if $\phi(t) = \max\{t, 0\}$ (ReLU activation) or $\phi(t) = t^2$, then a suboptimal local-min exists; if $\phi(t) = t^2 + 1$ or ϕ is strictly increasing, then there is no suboptimal local-min. When x and y are in high-dimensional space, what conditions guarantee the nonexistence of suboptimal local minima are still not fully understood, though partial progress has been made. In the next two sections, we discuss a few results for more general neural nets.

A major goal of theoretical research is to identify the critical factors of modern neural networks that contribute to successful training.

Bad local minima for ReLU networks

Due to the popularity of ReLU activation, a few works analyzed two-layer ReLU networks. For instance, [33] and [40] constructed suboptimal local minima for two-layer ReLU networks under different settings. The existence of suboptimal local minima for ReLU networks is not surprising, since Proposition 1 showed that, even for a single-neuron network with ReLU activation, a suboptimal local-min can exist. Nevertheless, a rigorous analysis for multineuron ReLU networks is nontrivial and requires other techniques. Due to space limitations, we do not review these results in detail here.

Does overparameterization eliminate bad local minima for smooth neurons?

One major result reviewed in the survey in [7] is that a wide one-hidden-layer network has no suboptimal local minima. At that time, researchers thought the assumption of "many hidden neurons" was restricted. Today, this assumption is considered rather reasonable; thus, it is worthwhile to revisit this classical result more carefully. Reference [7] did not cite the full result, and we cite the version from [43, Th. 3] in Claim 1.

Claim 1

Consider the problem $\min_{v\in\mathbb{R}^{1\times p}, W\in\mathbb{R}^{p\times d_x}} \Sigma_{i=1}^{n} (y_i - v\phi(Wx_i))^2$, *where* ϕ *is a sigmoid function. Assume the width* $p \geq n$ *and that there is one index k such that* $x_{ik} \neq x_{jk}, \forall i \neq j$. *Then, every local minimum is a global minimum [43, Th. 3].*

Unfortunately, it was recently found that this claim does not hold. A counterexample to this claim was given in [13]. A modification to this claim will be discussed later.

Cavity of the proof of Claim 1

To prove Claim 1, [43] first proved the function satisfies the following property (called *Property PT* for short).

Definition 2

Property PT: We say a function F satisfies Property PT if, starting from any point θ, *there exists an arbitrarily small perturbation such that, from the perturbed point* $\hat{\theta}$, *there exists a strictly decreasing path to a global minimum.*

Yu and Chen [43] claimed that Property PT implies the nonexistence of suboptimal local minima. This deduction contains a cavity, as demonstrated in Figure 2. This function contains at least one local-min (indicated by the red point). However, if starting from a suboptimal local-min (red point), after a small perturbation (the blue point), there is a strictly decreasing path (colored in blue) to the global-min. Therefore, even if the function satisfies Property PT, a suboptimal local-min can still exist.

Note that the loss function in Figure 2 does not use sigmoid activation; thus, this function can only demonstrate that "Property PT does not imply the nonexistence of suboptimal local-min," indicating a cavity of the proof of [43], but does not serve as a counterexample to the result of [43] (i.e., Claim 1). We will present a counterexample next.

The existence of a suboptimal local-min for arbitrarily wide networks

Reference [13] directly proved that Claim 1 does not hold by providing a counterexample.

Proposition 2

Let $n \geq 3$. *For a neural network with sigmoid activation and input data* $x_1, \cdots, x_n \in \mathbb{R}$, *where* $x_i \neq x_j$ *for all* $i \neq j$, *there exist output data* $y_1, \cdots, y_n$ *such that the empirical loss has a suboptimal local minimum.*

Besides the sigmoid activation, [13] also proved a stronger negative result that, for a large class of smooth activation functions, arbitrarily wide and deep networks, with generic input data x_i's with dimensions $d_x^2 + 3d_x/2 < n$, there exist output data y_i's such that suboptimal local minima exist. It is unknown whether allowing picking labels (e.g., label smoothing) can eliminate suboptimal local minima.

The absence of bad valleys and basins

Although suboptimal local minima can exist for wide neural networks, researchers indeed found that a large width is critical for good performance. Thus, one may expect that wide networks exhibit some nice geometrical properties. In this section, we review the results on such properties.

No spurious valley for increasing activations

Definition 3

A spurious valley is a connected component of a sublevel set $\{\theta : F(\theta) \leq c\}$ *that does not contain a global minimum of the loss* $F(\theta)$.

The nonexistence of a spurious valley guarantees the nonexistence of a suboptimal strict local-min. Although there may still exist suboptimal nonstrict local minima, the absence of a spurious valley ensures that, starting from any of these suboptimal local minima, there exists a nondecreasing path (not necessarily a strictly decreasing path) to a region with smaller loss [42].

Reference [42] proved that no spurious valley exists (implying no bad basin) for a one-hidden-layer network with "low intrinsic dimension." Reference [37] further proved that there is no spurious valley for wide deep neural networks where the last hidden layer has no fewer neurons than the number of samples, under a few assumptions on the activation functions. This is given in the following theorem.

Theorem 2

Suppose that an arbitrarily deep fully connected neural network $f_\theta(x)$ satisfies the following assumptions.

- *The activation function σ is strictly monotonic, and $\sigma(\mathbb{R}) = \mathbb{R}$.*
- *For any integer $p \geq 2$, there do not exist nonzero coefficients $(\lambda_i, a_i)_{i=1}^{p}$ with $a_i \neq a_j, \forall i \neq j$, such that $\sigma(x) = \Sigma_{i=1}^{p} \lambda_i \sigma(x - a_i)$ for every $x \in \mathbb{R}$.*
- *$d_L \geq n$.*
- *All of the training samples are distinct.*

Then, the empirical loss $F(\theta)$ has no spurious valleys.

No suboptimal basin for any continuous activations

The "no spurious valley" result in [39] holds for strictly increasing analytic activation functions, but it does not cover many nonsmooth or nonmonotone activations that are commonly applied in practice, such as leaky ReLU or swish. Reference [29] analyzed deep, overparameterized neural networks with any continuous activations. The result relies on a notion called *setwise strict local minimum*, defined here.

Definition 4

Setwise strict local minimum: We say a compact subset $X \in S$ is a strict local minimum of $f : S \to \mathbb{R}$ in the sense of sets if there exists $\varepsilon > 0$ such that, for all $x \in X$ and for all $y \in S \setminus X$ satisfying $\| x - y \|_2 \leq \varepsilon$, it holds that $f(x) < f(y)$.

Definition 4 generalizes the notion of the strict local-min from the sense of points to the sense of sets. Subsequently, we introduce the concept of suboptimal basin.

Definition 5

Suboptimal basin: A suboptimal basin of a function $f : S \to \mathbb{R}$ is a setwise strict local minimum that does not contain a global minimum of f.

A function that has no suboptimal basin may still have (pointwise) suboptimal local minima, which can only form flat areas called *plateaus*. We note that such plateaus cannot lie in the bottom of a suboptimal basin, as illustrated in Figure 3. Reference [29] proved that, for all deep neural networks where the last hidden layer is wider than the number of samples, the loss function has no suboptimal basin.

Theorem 3

Suppose that an arbitrarily deep fully connected neural network $f_\theta(x)$ satisfies the following assumptions.

- *There exists k such that $(x_i)_k \neq (x_j)_k, \forall i \neq j$, where $(x_i)_k$ indicates the kth entry of x_i.*
- *$d_L \geq n$.*
- *The activation σ_l is continuous: $l = 1, \cdots, L$.*

Assume the loss function $l(a, b)$ is convex with respect to b. Then, the empirical loss $F(\theta)$ defined in (2) has no suboptimal basin.

- *Remark 1*: The two theorems can be generalized to deep neural networks with one wide layer (not necessarily the widest); see, e.g., [29, Th. 2]. Due to space restrictions in this article, we do not review these results.
- *Remark 2*: "Suboptimal basin" is closely related to "spurious valley": every suboptimal basin must contain a spurious valley, but not vice versa. If a function has no spurious valley, then it does not have setwise local minima; the reverse is not true. Not every spurious valley is a suboptimal basin, because a spurious valley is not necessarily compact. Not every suboptimal basin is a spurious valley as well, since the latter has to be a subset of a sublevel set. The simplest statement about their relation is that "no spurious valley implies no suboptimal basin." Why are there two notions, "valley" and "basin"? Different "conclusions" (no spurious valley versus no setwise local minima) require a different set of assumptions (strictly increasing smooth neurons versus any continuous neuron); thus, the two results are currently not replaceable. It is an open question whether there exists a universal result that includes both results as special cases.

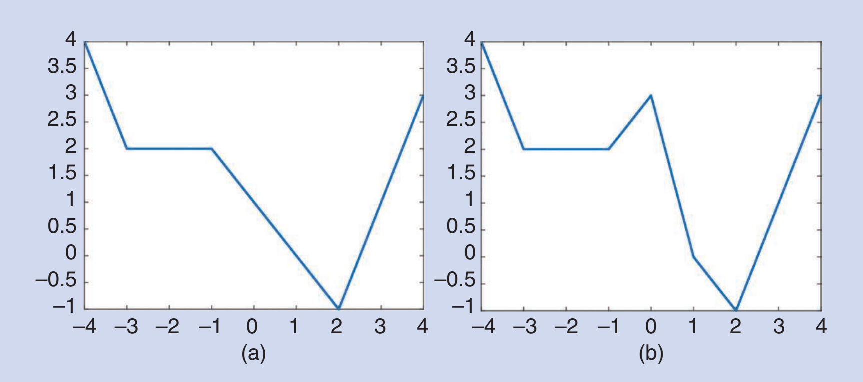


FIGURE 3. An example of (a) a function without a suboptimal basin and (b) a function with a suboptimal basin. Both functions have bad nonstrict local minima, consisting a plateau of $(-3, -1)$. Notice that the plateau in (b) lies in a suboptimal basin. (Source: reprinted from [29].)

- *Remark 3*: We implicitly assume that a global-min exists. In machine learning, a global-min may not exist, and only a global infimum exists; for simplicity of presentation, we do not add this extra degree of complication throughout the article.

Narrow networks

Previous results assume that the network width is large (at least n). Reference [29] presented a result showing $n-1$ neurons are not enough to eliminate suboptimal basins.

Proposition 3

For any n input data $x_1,\cdots,x_n \in \mathbb{R}$ with $x_i \neq x'_j, \forall i \neq j$, there exist n outputs $y_1,\cdots,y_n \in \mathbb{R}$ and a one-hidden-layer neural network with $n-1$ neurons such that the empirical loss $E(\cdot)$ has a bad strict local minimum.

This result, together with Theorem 3, demonstrates that adding enough neurons can eliminate suboptimal basins. Note that this result has a number of restrictions (e.g., special output data and special neurons), and a general condition for the existence of suboptimal basins requires more research.

Empirical explorations of the landscape

We have discussed a few theoretical results on wide neural nets. In this part, we discuss some empirical explorations, which reveal nontrivial properties of the landscape. Some parts are accompanied by theoretical results; however, the main motivation of the whole section is mostly empirical rather than proving theorems. Some of the findings are consistent with the theoretical results we discussed before, and some of the findings call for more in-depth theoretical understanding.

Visualization of the landscape

Landscape is a geometrical subject; thus, visualization of the landscape is very useful for understanding. For 1D or 2D functions, it is common to draw the plot $(w,f(w))$ for w in an interval (1D) or a box (2D) and draw the contour $\{w: f(w)=c\}$ for various values of c. However, visualizing objects in a high-dimensional space is difficult in general. A number of dimensionality-reduction schemes have been suggested to partially visualize the landscape of neural networks.

In [19], the authors consider the straight line between two points θ_1 and θ_2 and evaluate the function on the line segment connecting them. Consider an algorithm that generates a sequence of points $\theta^k = \mathcal{A}(\theta^{k-1}), k=1,2,\ldots,T$, where T is the total number of iterations. One example is the GD algorithm $\mathcal{A}(\theta)=\theta-\eta\nabla F(\theta)$ for a certain learning rate η; instead of GD, Goodfellow et al. [19] tested the popular stochastic GD (SGD). They picked a random initial point $\theta^0=\theta_1$, picked the converged solution $\theta^T=\theta_2$, and drew the plot of the function $F_{[\theta_1,\theta_2]}(\alpha), \alpha \in [0,1]$, where

$$F_{[\theta_1,\theta_2]}(\alpha) \triangleq F(\alpha\theta_1+(1-\alpha)\theta_2).$$

They showed empirically that the function value is decreasing from $\alpha=0$ to $\alpha=1$ (except a small bump near the initial point sometimes). This phenomenon will naturally occur when optimizing a convex function, but why this happens in neural network training is largely unknown. We present their finding as the following formal conjecture.

Conjecture 1

Consider a random initial point θ^0 and suppose SGD generates a sequence $\theta^k, k=1,\ldots$. Further, assume the limit $\lim_{k=1}^{\infty}\theta^k=\theta^*$ exists. Then, under certain conditions on the neural nets, $F_{[\theta^0,\theta^*]}(\alpha)$ is a strictly decreasing function in the interval $\alpha \in [0,1]$.

Reference [30] visualized the landscape by projecting it onto a 2D space. More specifically, a center point θ_0 and two vectors v_1 and v_2 are picked, and the function values $F(\theta_0+\alpha v_1+\beta v_2)$ are plotted for $\alpha,\beta \in [-1,1]$. The basis vectors v_1, v_2 are chosen by a certain special scaling (called *filter normalization*) of random Gaussian vectors. It was empirically shown in [30] that the 2D landscape is highly correlated with the trainability of the networks, as demonstrated in Figures 4 and 5: deep networks without skip connections are hard to train, and their 2D landscapes have "dramatic nonconvexities"; in contrast, deep residual neural networks (ResNets) and DenseNet are easy to train in practice, and they, indeed, have convex contours. To help readers understand the empirical findings, we present Conjecture 2.

Conjecture 2

Consider a global minimum θ^* and two vectors v_1, v_2 drawn by a certain rule (e.g., Gaussian distribution). Define the function

$$G_{v_1,v_2}(\alpha,\beta)=F(\theta^*+\alpha v_1+\beta v_2), \alpha,\beta \in \mathbb{R}.$$

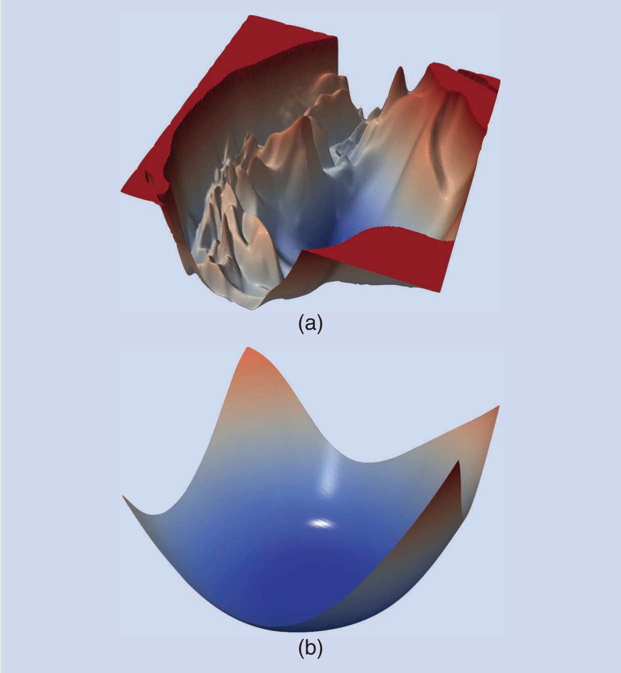
Then, for standard neural nets with width above a threshold c_1 or ResNets with width above a threshold $c_2 < c_1, G_{v_1,v_2}(\alpha,\beta)$ has no suboptimal basins. In addition, for standard neural nets with width below a threshold $c'_1 < c_1$, $G_{v_1,v_2}(\alpha,\beta)$ has many basins.

Theorem 3 and Proposition 3 discussed earlier (appearing in [29]) show a distinction between narrow and wide networks and, thus, have a similar flavor to Conjecture 2. Nevertheless, it is unknown whether the original version of Conjecture 2 can be proven.

Mode connectivity

In this section, we present another interesting empirical finding, supported by some theoretical results. Draxler et al. [14] and Garipov et al. [18] empirically found that two global minima can be connected by an (almost) equal-value path. This means that the "modes" (meaning different global minima) are connected via an equal-value path, which explains the terminology *mode connectivity*. We provide a formal description as follows.

Define $M(v_1,v_2,v_3)$ as the linear space spanned by $v_i, i=1,2,3$ for any three vectors v_1, v_2, v_3 (assuming they are linearly independent). Reference [18] generated Figure 6(a) as


(a)

(b)

FIGURE 4. A visualization of the landscape by projecting onto a 2D space. (a) A variant of ResNet-110 without skip connection. (b) A dense neural network with 121 layers. (Source: reprinted from [30]; used with permission.)

follows: first, train a standard 164-layer ResNet to find three solutions $\theta_i^*, i = 1,2,3$ by starting from three random initial points; second, define a function $f(s,t) = F(s\theta_1 + t\theta_2 + (1-s-t)\theta_3)$, where $s,t \in \mathbb{R}$; third, draw the contour of the function $f(s, t)$ for s, t in certain intervals. Note that we can interpret the three solutions as three global minima, even though they are not exact global minima. The plot shows that the three solutions lie at the bottom of three basins; thus, we can make the following conjecture.

Conjecture 3

Suppose $\theta_i^*, i = 1,2,3$ are three global minima of F. Then, in any continuous path in the plane $M(\theta_1^*, \theta_2^*, \theta_3^*)$ that connects θ_1^* and θ_2^*, the maximum of F is strictly larger than $F(\theta_1^*)$.

To understand whether θ_1^* and θ_2^* are connected via some equal-value path, we can search over the space of paths. Figure 6(b) empirically shows that there exists a simple path that connects two global minima.

Conjecture 4

Suppose θ_1^* and θ_2^* are two global minima of F. There exists θ_0 such that the following holds: there exists a continuous path in the plane $M(\theta_1^*, \theta_2^*, \theta_0)$ that connects θ_1^* and θ_2^* and passes θ_0 along which the value of F is constant.

In optimization language, mode connectivity means that the sublevel set $\{\theta \mid F(\theta) \le F^*\}$, which is the same as $\{\theta \mid F(\theta) = F^*\}$, is connected, where F^* is the global minimal value. These findings are partially motivated by Freeman and Bruna [17], who proved a stronger property that the sublevel set $\{\theta \mid F(\theta) \le c\}$ is connected for any c for deep linear networks and one-hidden-layer ultrawide ReLU networks. Kuditipudi et al. [27] and Nguyen [37] provided a theoretical justification for this phenomenon; due to space limitations, we do not discuss their results in detail.

Now, we describe the empirical method used in [14] and [18] to verify mode connectivity. The goal is to find an equal-value path connecting two global minima. In practice, it is hard to find exact global minima; thus, a reasonable replacement is to train a neural net to find two different solutions $\theta_i^*, i = 1,2$ by starting from two random initial points. To find a path P connecting two points θ_1^* and θ_2^* with "equal value," these works use an optimization problem: find a path with the lowest "energy," where the *energy* can be defined in different ways. Reference [14] minimizes the "infinity norm" of the path P, i.e., solving $\min_{P \text{ from } \theta_1^* \text{ to } \theta_2^*} \max_{\theta \in P} F(\theta)$, and [18] minimizes the "ℓ_1 norm" of the path, i.e., solving $\min_{P \text{ from } \theta_1^* \text{ to } \theta_2^*} \mathbb{E}_\theta F(\theta)$, where θ is drawn from a certain random distribution on the path P.

We briefly discuss the practical tricks used in [18]. There are a huge number of continuous paths from θ_1^* to θ_2^*. To restrict the search space of the paths, the authors consider a subclass of paths, such as the class of Bezier curves $\theta_\psi(t) = (1-t)^2\theta_1^* + 2t(1-t)\psi + t^2\theta_2^*, t \in [0,1]$, where ψ is any parameter. Then, they use SGD to solve

$$\min_{\psi} E_{t \sim U[0,1]} F(\theta_\psi(t)).$$

The result is illustrated in Figure 6(b). The choice of Bezier curve is arbitrary, and they also report the results of using other curves.

Mode connectivity is not only an interesting geometrical finding; it has practical implications as well. For example, mode connectivity implies that, once we find two global minima, there is likely to be a connected path between the two minima. This provides an opportunity for searching for better minima that yield lower test errors. In [18], such a technique has been proposed.

Saddle points or local minima

An influential early article in the recent wave of landscape analysis by Dauphin et al. [12] studied which points caused training difficulties. It advocated the hypothesis that saddle points instead of local minima are a major issue for neural network training. The underlying logic is the following: under the conjecture that most local minima are close to global minima, if an algorithm gets stuck at a point with a large error, then it is likely to be a saddle point instead of a local-min. This claim is one of the motivations for many later works on escaping saddle points (see the "Escaping Saddle Points" section). Whether saddle points or local minima are a more severe issue remains an interesting question.

Eliminating bad local minima for nonlinear networks

In the previous section, we found that eliminating suboptimal local minima globally is very difficult; thus, we resort to more complicated concepts, such as spurious valleys. In this section,

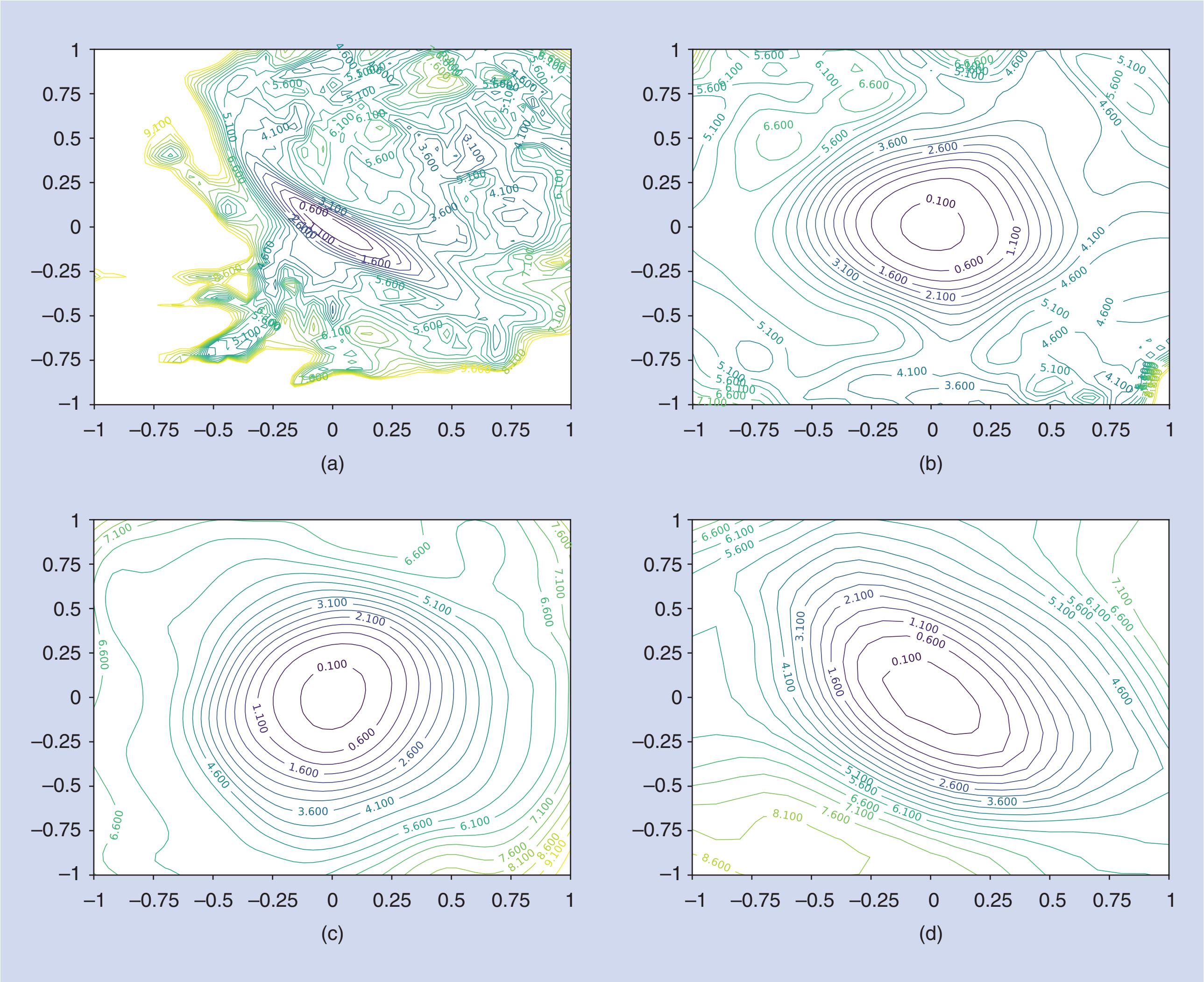


FIGURE 5. A visualization of the landscape of ResNet-56 with different widths: the widths are (a) 1, (b) 2, (c) 4, and (d) 8 times as large as the original ResNet-56. The test errors are (a) 13.31%, (b) 10.26%, (c) 9.69%, and (d) 8.70%, respectively. (Source: reprinted from [30]; used with permission.)

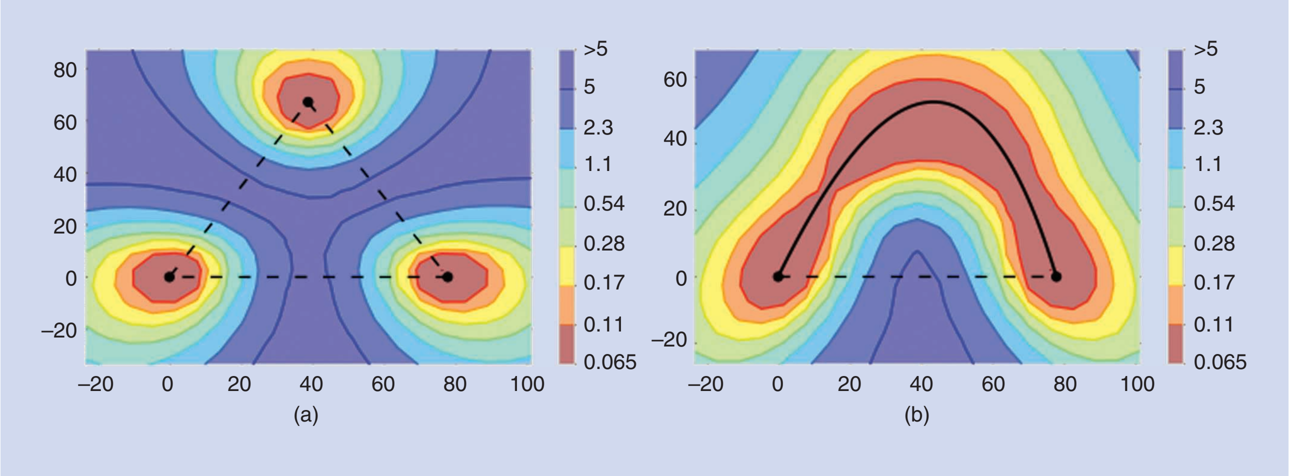


FIGURE 6. The mode connectivity. These are the contours of the loss of a 164-layer ResNet trained on CIFAR100 as a function of the network weights in a 2D subspace. This subspace is spanned by the three points θ_1^*, θ_2^* (which are fixed), and ψ (which can be changed). Here, θ_1^*, θ_2^* are two solutions (likely local optima) found by training the network from two independent initial points. (a) ψ is another solution found by training from another initial point, and we can see the barriers between the three minima. (b) ψ is found by solving a problem, and we can see a quadratic Bezier curve connecting the two optima along a path of near-constant loss. (Source: reprinted from [18]; used with permission.)

we follow a different path: we still try to eliminate bad local minima, but we allow the modification of other parts of the game. First, we discuss the results that eliminate bad local minima in a subset but not the whole space. Second, we show how to force all local minima to fall into a subset so that no bad local-min exists. Third, we discuss the limitation of eliminating bad minima and describe a stronger landscape property and a result on it.

Eliminating bad local minima in a subset

Local minima with full-rank postactivation matrices

Reference [38, Th. 3.4 and Th. 3.8] provided conditions for the absence of local minima with a certain full-rank condition. Here, we present a version in [29] that is different from [38] Define $z_0(x) = x$ and $z_k(x) = \phi(W_{k-1} z_{k-1}(x) + b_k), k = 1, 2, \ldots, L.$ Then, we can write $f_w(x) = W_L Z_L(x)$. Let $X = (x_1, \ldots, x_n)$ and let $Z_L(X) = (z_L(x_1), \ldots, z_L(x_n))$.

Claim 2

Define $\mathcal{W}_{\text{L,full}} = \{\theta : Z_L(X) \text{ is full rank}\}$. Every local minimum of $F(\theta)$ in the set $\mathcal{W}_{\text{L,full}}$ is a global minimum.

Claim 3

Suppose the kth-order derivatives of the activation function $\sigma^{(k)}(0)$ are nonzero, for $k = 0, 1, 2, \ldots, n-1$. Then, the set $\mathcal{W}_{\text{L,full}}$ is dense.

Claim 2 is of interest on its own since it is rather simple. The results discussed in the "The Absence of Bad Valleys and Basins" section can be viewed as more modern versions (with no restriction to a subset).

Local minima with full-rank neural tangent kernels

There is another simple result of a similar flavor. For simplicity of presentation, we assume $d_y = 1$. Define

$$G(\theta) = \left(\frac{\partial f_\theta(x_1)}{\partial \theta}, \ldots, \frac{\partial f_\theta(x_n)}{\partial \theta} \right) \in \mathbb{R}^{P \times n}, \tag{4}$$

where P is the number of parameters, and define the neural tangent kernel (NTK) as

$$K(\theta) = G(\theta)^T G(\theta). \tag{5}$$

Claim 4

Suppose $d_y = 1$ and $\ell(a, b) = (a-b)^2$. Define $\mathcal{W}_{\text{NTK}} = \{\theta : K(\theta) \text{ is full rank}\}$. Every critical point θ^* of $F(\theta)$ in the set $\mathcal{W}_{\text{NTK,full}}$ is a global minimum.

Proof

Let $e_i^* = f_{\theta^*}(x_i) - y_i$ and $e^* = (e_1^*; \cdots; e_n^*) \in \mathbb{R}^{n \times 1}$. Since $F(\theta) = 1/n \Sigma_{i=1}^n (y_i - f_\theta(x_i))^2$, we have

$$\frac{dF(\theta^*)}{d\theta} = \frac{2}{n} G(\theta^*) e^*.$$

If $dF(\theta^*)/d\theta = 0$ and $G(\theta^*)$ is full rank, we have $e^* = 0$ and $F(\theta^*) = 0$, implying θ^* is a global-min.

The full rankness of $G(\theta)$ is equivalent to the full rankness of $K(\theta)$; we do not need $K(\theta)$ here, but we still define $K(\theta)$ since it is critical in NTK theory. Despite its simplicity, Claim 4 can be viewed as the foundation of the NTK theory we discuss later.

Local minima with an inactive neuron

The idea of considering local minima with special structure dates back to at least the classical work on Burer-Monteiro factorization [8]. It showed that, for a certain class of nonconvex matrix problems, a local-min with a zero column must be a global-min. This result is not directly related to neural nets, but it indicated an interesting direction.

Reference [20] analyzed a two-layer neural network $W_2 \phi(W_1 x)$ with positive homogeneous activations (e.g., ReLU, linear). It proved that a local-min with one inactive neuron is a global-min. (A formal result is somewhat technical and omitted here.) Note that the optimization variables are the two weight matrices W_1 and W_2; thus, "an inactive neuron" means that there is an index j such that the jth rows of W_1 and $W_2^\top$ are both zero. This can be viewed as a variant of the result of [8].

Comments

Eliminating bad local minima in a subset itself is of limited interest, since an algorithm may or may not stay in this subset. Extra techniques are required to make these results more interesting. The three results we presented in this section are, indeed, extended to three stronger results (Theorem 3, NTK theory [22], and Theorem 5, respectively). Again, we present them here since they are simple and provide some insight.

Making modifications to eliminate bad local minima

Eliminating bad local minima by ensuring an inactive neuron

Reference [32] provides two modifications of the system, each of which can ensure no bad local-min exists, for binary classification. In the first modification, for any deep neural network, [32] added a special neuron (e.g., exponential) from input to output and a quadratic regularizer on its weight. The second modification is to use a special neuron (e.g., exponential) at each layer and add regularizers for the weights connected to these special neurons. The two modifications are demonstrated in Figure 7.

Next, we present the result for the first modification [32, Th. 1]. Assume that there exists a $\boldsymbol{\theta}$ such that the neural net f_θ can correctly classify all samples in the data set. Now, we add an exponential neuron to the architecture and have a modified function $\tilde{f}(x; \tilde{\boldsymbol{\theta}}) = f_{\boldsymbol{\theta}}(x) + a \exp(w^\top x + b)$. For the logistic loss function $\ell(y, z) = \log_2(1 + e^{-yz})$, we consider a modified loss function

$$\tilde{L}_n(\tilde{\boldsymbol{\theta}}) = \sum_{i=1}^n \ell(y_i, \tilde{f}(x; \tilde{\boldsymbol{\theta}})) + \frac{\lambda a^2}{2}. \tag{6}$$

The original loss function is defined as

$$L_n(\tilde{\boldsymbol{\theta}}) = \sum_{i=1}^{n} \ell(y_i, f_{\boldsymbol{\theta}}(x_i)).$$

Theorem 4

Under the presented settings, we have the following ([32, Th. 1]).

1) *The function $\tilde{L}_n(\tilde{\boldsymbol{\theta}})$ has at least one local minimum.*
2) *At every local minimum, $a = 0$.*
3) *Assume that $\tilde{\boldsymbol{\theta}}^* = (\boldsymbol{\theta}^*, a^*, \boldsymbol{w}^*, b^*)$ is a local minimum of $\tilde{L}_n(\tilde{\boldsymbol{\theta}})$; then, $\tilde{\boldsymbol{\theta}}^*$ is a global minimum of $\tilde{L}_n(\tilde{\boldsymbol{\theta}})$. Furthermore, $\boldsymbol{\theta}^*$ achieves the minimum loss value on the data set $\mathcal{D}$, i.e., $\boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta}} L_n(\boldsymbol{\theta})$.*

The proof consists of two steps. Liang et al. [32] first showed that, at any critical point of the loss function, the exponential neuron is always inactive. This trick allows us to consider the local-min in a subset (the topic of the previous section). Then, [32] proved that a local-min with an inactive neuron is a global-min.

Extension to multiclass classification and regression

Reference [25] extended [32] (the first modification) to the multiclass classification tasks. Similar to the construction proposed by [32], it added an exponential neuron on the output of the neural network for each class and added an ℓ_2 regularizer for the parameters of all exponential neurons. The high-level proof ideas adopted those of [32] (though with some technical differences). It first showed that, at every local-min of the empirical loss function, all exponential neurons are inactive. Then, it showed that a local-min with these neurons inactive must be a global-min.

The limitations of eliminating bad local minima

References [25] and [32] showed that it is not difficult to prove every local-min is a global-min as long as small modifications can be made. However, [25] argued that there are simple examples where, on the modified landscape, there are new paths leading the original local-min to infinity, and, thus, a descent algorithm might diverge to infinity. We remark that most works on the landscape analysis of neural networks mentioned previously do not explicitly eliminate the possibility that a descent algorithm will diverge to infinity. For instance, for a three-layer 1D linear network problem $\min_{x,y,z\in\mathbb{R}}(xyz-1)^2$, although every no suboptimal local-min exists according to [24], there is a sequence $(x_k, y_k, z_k) = (-1/k, \sqrt{k}, 1/k)$ diverging to infinity while the function values are decreasing and converging to one, which is clearly a suboptimal value. This shows that a "decreasing path" to infinity exists even for linear neural networks.

Eliminating bad local minima and decreasing paths to infinity

A natural question is, then, whether one can further eliminate the possibility of decreasing paths to infinity as well as suboptimal local minima. All of the results we discussed so far cannot satisfy both properties together. Some results prove no suboptimal local-min [24], [25], [32], [33], but their loss functions may have a decreasing path to infinity.

Reference [34] provides a positive answer to the question; it considers overparameterized neural nets with arbitrary depth. For simplicity of presentation, we state the authors' result for a one-hidden-layer network. This network can be expressed by $f(x;\boldsymbol{\theta}) = \Sigma_{j=1}^{m} a_j \mathrm{ReQU}(\boldsymbol{w}_j^\top x + b_j)$, where the activation function is $\mathrm{ReQU}(z) = [\max\{z, 0\}]^2$ and θ denotes the collection of all parameters. Here ReQU means "rectified quadratic units." Suppose the loss function is logistic: $\ell(y;z) = \log(1 + e^{-yz})$. Furthermore, reference [34] assumes that the data points $x_1, \ldots, x_n$ are distinct. The loss to minimize is

$$F(\boldsymbol{\theta}) = \sum_{i=1}^{n} \ell(y_i; f(x_i;\boldsymbol{\theta})) + \frac{1}{3}\sum_{j=1}^{m} \lambda_j\left[|a_j|^3 + 2\left(\|w_j\|_2^2 + b_j^2\right)^{3/2}\right], \tag{7}$$

where all regularizer coefficients λ_js are positive numbers, and the vector $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_m)$ consists of all regularizer coefficients. Reference [34] shows that, if the network size is larger than the data set size, i.e., $m \geq n+1$, and the regularizer coefficient vector $\boldsymbol{\lambda}$ is chosen in a specific way, then every local-min

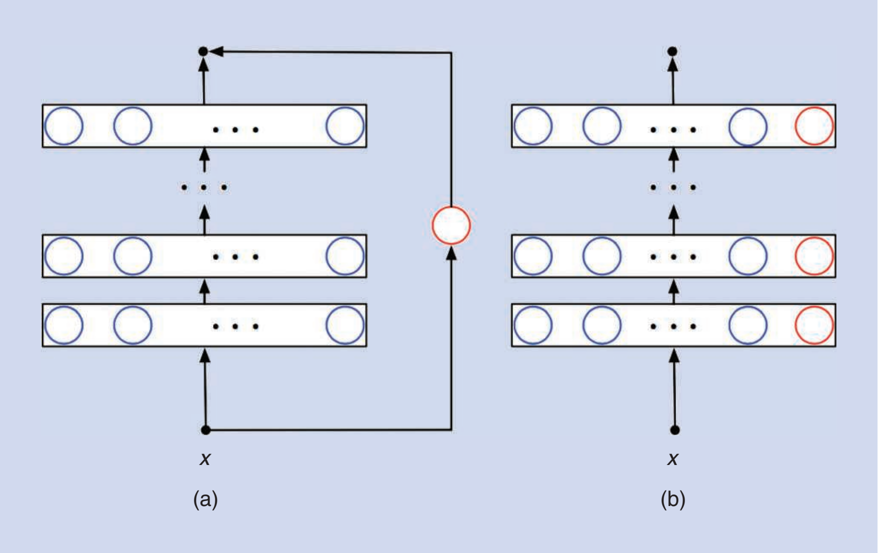


FIGURE 7. The modifications. (a) A special neuron (e.g., exponential) is added from input to output. (b) The architecture is a regular fully connected neural network, and, at each layer, there is a special neuron (e.g., exponential). (Source: reprinted from [32]; used with permission.)

achieves zero training error. In this result, we use a standard notion called *coercive*: we say F is a coercive function if and only if $\lim_{\|\theta\|\to\infty} F(\theta)=\infty$; thus, a coercive function has no decreasing path to infinity.

Theorem 5

Let $m \geq n+1$. There exist a $\lambda_0=\lambda_0(\mathcal{D},\ell)>0$ and a zero measure set $C \subset \mathbb{R}^m$ such that, for any $\boldsymbol{\lambda} \in (0,\lambda_0)^m \setminus C$, both of the following statements are true:

1) The empirical loss $F(\boldsymbol{\theta})$ is coercive.
2) Every local-min $\boldsymbol{\theta}^*$ of the loss $F(\boldsymbol{\theta})$ is a global-min of $F(\boldsymbol{\theta})$ and achieves zero training error.

Remark

When all data points are distinct and the size of the ReQU network (i.e., the network with ReQU activation functions) is larger than the size of the data set, it is straightforward to show that every sample in the data set can be correctly classified by the neural network. In other words, there exists θ^* such that $F(\theta^*)=0$. This fact is commonly known as *overparameterization implies interpolation.*

The limitation of the result is that it considers a special neuron called *ReQU.* Nevertheless, this result at least shows the possibility of achieving both "no bad local-min" and "no decreasing path to infinity."

An alternative two-step approach (used in NTK-type analysis) is the following: 1) prove that, in a subset of the parameter space, every local-min is a global-min and 2) prove that, under certain conditions, the iterates always stay in this subset.

Algorithmic analysis

As mentioned in the "Introduction" section, although algorithmic analysis is very important and closely related to the optimization landscape, it is not the focus of this article. Nevertheless, we briefly discuss some related results and concepts for a better big picture.

Escaping saddle points

If no suboptimal local-min exists (e.g., as proven in [34], discussed in the "Eliminating Bad Local Minima and Decreasing the Paths to Infinity" section), then a local search algorithm is likely to converge to a saddle point or a global-min. A rigorous analysis may need to prove that a local search algorithm can escape saddle points. Theoreticians distinguish second-order saddle points and high-order saddle points (see [3]). It is well known that second-order strict saddle points (critical points whose Hessians have a negative eigenvalue) can be escaped by GD [28], though, in the worst case, it takes exponential time [16].

Other works (e.g., [23]) show that noisy GD can escape second-order strict saddle points in polynomial time. Reference [3] devised a polynomial time algorithm for escaping third-order saddle points and also showed that escaping fourth-order or higher-order saddle points is NP-hard. Therefore, to prove a rigorous convergence result of GD with a polynomial time convergence bound, just proving "no suboptimal local minima" may not be enough.

Nevertheless, NP-hardness is proven for special nonconvex problems, not neural net problems. Thus, by exploring the structure of neural nets, it is still possible to prove polynomial time convergence with the presence of high-order saddle points. However, the convergence analysis with the presence of saddle points is mostly done for generic nonconvex problems and rarely studied for neural networks (note that the characterization of saddle points is common in neural net articles, e.g., [32], [44], and [45], but no convergence analysis follows). It requires further research to rigorously prove the convergence of GD or SGD to the global-min for the neural nets discussed in the "Does Overparameterization Eliminate Bad Local Minima for Smooth Neurons?" section.

Note that, for a landscape with no spurious valley or no bad basin (discussed in the "The Absence of Bad Valleys and Basins" section), a convergence analysis of GD is even harder. We suspect that the absence of a basin is enough for a noisy variant of GD to converge to the global-min (perhaps utilizing some structural property of the neural nets). Such a convergence result requires more research.

There is another perspective [26] on how SGD can help converge to the global-min: running SGD on a nonconvex function can be seen as running GD on a "smoothed function," which is the convolution of the original function and a noise. The bad local minima of the original function may disappear in the "smoothed landscape." This potentially explains why SGD can escape bad local minima in the optimization of nonconvex function, such as neural networks. Nevertheless, [26] only studied special shallow networks. A general analysis of deep neural nets using this perspective is still missing.

Algorithmic analysis for ultrawide networks

In the "Escaping Saddle Points" section, we discussed a two-step approach of proving global convergence: first, prove the neural net landscape exhibits certain property; second, apply a generic convergence analysis for a function with this property. An alternative two-step approach (used in NTK-type analysis) is the following: 1) prove that, in a subset of the parameter space, every local-min is a global-min and 2) prove that, under certain conditions, the iterates always stay in this subset.

Recall that the NTK $K(\theta)=G(\theta)^T G(\theta)$, where $G(\theta)=(\partial f_\theta(x_1)/\partial\theta, \ldots, \partial f_\theta(x_n)/\partial\theta)$, assuming $d_y=1$. According to Claim 4, if the NTK is always full rank (more precisely, the singular value has a positive lower bound) for all iterates of GD, and, further, GD converges to a critical point, then GD converges to a global-min. Recent works [5], [22] prove that, with infinite width [or poly(n) neurons per layer], the NTK $K(\theta)$ stays full rank along the trajectory of GD with a random initialization, thus finishing the convergence proof.

Note that we present a simplified framework to demonstrate the key idea of how [5] and [22] prove global convergence.

However, the power of the NTK framework is proving not just convergence but also the linear convergence rate and even the generalization error bound. These aspects are beyond the scope of this article, so they are not discussed in detail here. Around the same time as [22], references [1], [2], [4], [15], [31], and [46] also proved the global convergence of GD under similar "ultrawide" conditions; we skip the details of those results here.

Despite the strong conclusions (convergence, convergence rate, and so on), the assumption of a large width is not satisfied by practical neural nets. Nevertheless, a more important aspect is the theoretical insight. An intuition of NTK theory is that, for extremely wide networks, the weights have little change during the whole training procedure, and, hence, the model behaves as its linearization around the initialization. However, [10] showed by experiments that the dynamics of the linearized networks are different from the practical training dynamics; thus, the theory of NTK may be not enough to fully explain the practical training.

When the theory does not fully match practice, we do not necessarily have to modify theory; we can also modify the practical system (as mentioned in Question 3 in the "Introduction" section). Jacot et al. [22] suggested a new system of using kernel GD to solve machine learning problems, where the kernel is the explicitly computed NTK. This essentially reduces a complex, multilayer, nonlinear network to a simple linear model. We stress that this is a new method and is different from practical training. Arora et al. [5] performed precise computation using this new system and reported promising results on image classification.

NTK and other convergence analyses of neural nets represent a very active area of research. It probably requires a whole article to fully describe the recent advances. The focus of this article is on the geometric side of neural net problems, as mentioned in the "Introduction" section; thus, we do not describe these works in more detail.

Conclusions

In this article, we reviewed recent progress on the understanding of the global landscape of neural networks. We addressed various empirical findings on the landscape and, also, many theoretical results. We first explored the results on deep linear networks that no bad local-min exists. We then discussed why a classical claim on "no bad local-min" for overparameterized networks fails to hold, and we showed that a more rigorous claim should be no spurious valley (or no bad basin). We analyzed how to perturb the loss functions to eliminate bad local minima, the limitation of "no bad local-min," and how to obtain a stronger landscape property. Finally, we briefly examined the existing convergence analysis and its challenges. Note that in this article we do not discuss the relation between the landscape and generalization error (e.g., [50], [51], and [52]) and do not discuss important training tricks like initialization and batch normalization; see [13] for an overview of these topics.

While the progress is encouraging, there are still many mysteries on the landscape of neural nets, especially the link between theory and practice. For instance, the benefit of width is still not fully understood. How to leverage the insight obtained from the theory to design better methods/architectures is also an interesting question.

While the progress is encouraging, there are still many mysteries on the landscape of neural nets, especially the link between theory and practice.

Authors

Ruoyu Sun (ruoyus@illinois.edu) received his B.S. degree in mathematics from Peking University, China, and his Ph.D. degree in electrical engineering from the University of Minnesota. He is an assistant professor in the Department of Industrial and Enterprise Systems Engineering and affiliated with the Coordinated Science Lab and Department of Electrical and Computer Engineering, the University of Illinois Urbana-Champaign (UIUC). Before joining UIUC, he was a visiting research scientist at Facebook Artificial Intelligence Research and was a postdoctoral researcher at Stanford University. He won second place in the Institute for Operations Research and Management Sciences (INFORMS) George Nicholson student paper competition and honorable mention in the INFORMS Optimization Society student paper competition. His current research interests lie in optimization and machine learning, especially deep learning and large-scale optimization.

Dawei Li (dawei2@illinois.edu) received his B.Sc. degree in applied mathematics from Peking University, China, in 2017. He is currently working toward his Ph.D. degree in the Department of Industrial and Enterprise Systems Engineering, University of Illinois Urbana-Champaign. His research interests include optimization and machine learning theory.

Shiyu Liang (sliang26@illinois.edu) is a Ph.D. student at the University of Illinois Urbana-Champaign (UIUC), advised by Prof. Rayadurgam Srikant. Before joining UIUC, he graduated from Shanghai Jiao Tong University, China, where he spent three years in Prof. Xinbing Wang's research group. His research interests include the theories and applications of deep learning.

Tian Ding (tianding@link.cuhk.edu.hk) received his B.Eng. degree in automation from Tsinghua University, China, in 2014, and his Ph.D. degree in information engineering from the Chinese University of Hong Kong in 2019. His research interests include wireless communications, optimization, and deep learning.

Rayadurgam Srikant (rsrikant@illinois.edu) is the codirector of the C3.ai Digital Transformation Institute and the Fredric G. and Elizabeth H. Nearing Professor in the Department of Electrical and Computer Engineering and the Coordinated Science Lab at the University of Illinois Urbana-Champaign. He was the recipient of the 2019 IEEE Koji Kobayashi Computers and Communications Award and has received several best paper awards, including the 2015 INFOCOM Best Paper Award and the 2017 Applied Probability Society Best Publication Award.

He was the editor-in-chief of *IEEE/ACM Transactions on Networking* from 2013 to 2017. His research interests include machine learning, applied probability, and communication networks. He is a Fellow of the IEEE.

References

[1] Z. Allen-Zhu, Y. Li, and Z. Song, "A convergence theory for deep learning via over-parameterization," in *Proc. Int. Conf. Machine Learning*, 2019, pp. 242–252.

[2] Z. Allen-Zhu, Y. Li, and Z. Song, "On the convergence rate of training recurrent neural networks," in *Proc. Advances Neural Information Processing Systems*, 2019, pp. 6673–6685.

[3] A. Anandkumar and R. Ge, "Efficient approaches for escaping higher order saddle points in non-convex optimization," in *Proc. Conf. Learning Theory*, 2016, pp. 81–102.

[4] S. Arora, S. Du, W. Hu, Z. Li, and R. Wang, "Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks," in *Proc. Int. Conf. Machine Learning*, 2019, pp. 322–332.

[5] S. Arora, S. S. Du, W. Hu, Z. Li, R. R. Salakhutdinov, and R. Wang, "On exact computation with an infinitely wide neural net," in *Proc. Advances Neural Information Processing Systems*, 2019, pp. 8139–8148.

[6] P. Baldi and K. Hornik, "Neural networks and principal component analysis: Learning from examples without local minima," *Neural Netw.*, vol. 2, no. 1, pp. 53–58, 1989. doi: 10.1016/0893-6080(89)90014-2.

[7] M. Bianchini and M. Gori, "Optimal learning in artificial neural networks: A review of theoretical results," *Neurocomputing*, vol. 13, nos. 2–4, pp. 313–346, 1996. doi: 10.1016/0925-2312(95)00032-1.

[8] S. Burer and R. D. Monteiro, "A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization," *Math. Program.*, vol. 95, no. 2, pp. 329–357, 2003. doi: 10.1007/s10107-002-0352-8.

[9] Y. Chi, Y. M. Lu, and Y. Chen, "Nonconvex optimization meets low-rank matrix factorization: An overview," *IEEE Trans. Signal Process.*, vol. 67, no. 20, pp. 5239–5269, 2019. doi: 10.1109/TSP.2019.2937282.

[10] L. Chizat, E. Oyallon, and F. Bach, "On lazy training in differentiable programming," in *Proc. Advances Neural Information Processing Systems*, 2019, pp. 2933–2943.

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

[12] Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio, "Identifying and attacking the saddle point problem in high-dimensional non-convex optimization," in *Proc. Advances Neural Information Processing Systems*, 2014, pp. 2933–2941. doi: 10.5555/2969033.2969154.

[13] T. Ding, D. Li, and R. Sun, Sub-optimal local minima exist for almost all over-parameterized neural networks. 2019. [Online]. Available: arXiv:1911.01413

[14] F. Draxler, K. Veschgini, M. Salmhofer, and F. Hamprecht, "Essentially no barriers in neural network energy landscape," in *Proc. Int. Conf. Machine Learning*, 2018, pp. 1309–1318.

[15] S. Du, J. Lee, H. Li, L. Wang, and X. Zhai, "Gradient descent finds global minima of deep neural networks," in *Proc. Int. Conf. Machine Learning*, 2019, pp. 1675–1685.

[16] S. Du, J. Lee, Y. Tian, A. Singh, and B. Poczos, "Gradient descent learns one-hidden-layer CNN: Don't be afraid of spurious local minima," in *Proc. Int. Conf. Machine Learning*, 2018, pp. 1339–1348.

[17] C. D. Freeman and J. Bruna, "Topology and geometry of half-rectified network optimization," in *Proc. Int. Conf. Learning Representations*, 2017.

[18] T. Garipov, P. Izmailov, D. Podoprikhin, D. P. Vetrov, and A. G. Wilson, "Loss surfaces, mode connectivity, and fast ensembling of DNNs," in *Proc. Advances Neural Information Processing Systems*, 2018, pp. 8789–8798.

[19] I. J. Goodfellow, O. Vinyals, and A. M. Saxe, Qualitatively characterizing neural network optimization problems. 2014. [Online]. Available: arXiv:1412.6544

[20] B. D. Haeffele and R. Vidal, "Global optimality in neural network training," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2017, pp. 4390–4398. doi: 10.1109/CVPR.2017.467.

[21] S. Han, H. Mao, and W. J. Dally, Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. 2015. [Online]. Available: arXiv:1510.00149

[22] A. Jacot, F. Gabriel, and C. Hongler, "Neural tangent kernel: Convergence and generalization in neural networks," in *Proc. Advances Neural Information Processing Systems*, 2018, pp. 8571–8580.

[23] C. Jin, R. Ge, P. Netrapalli, S. M. Kakade, and M. I. Jordan, "How to escape saddle points efficiently," in *Proc. 34th Int. Conf. Machine Learning*, 2017, pp. 1724–1732.

[24] K. Kawaguchi, "Deep learning without poor local minima," in *Proc. Advances Neural Information Processing Systems*, 2016, pp. 586–594.

[25] K. Kawaguchi and L. P. Kaelbling, Elimination of all bad local minima in deep learning. 2019. [Online]. Available: arXiv:1901.00279

[26] B. Kleinberg, Y. Li, and Y. Yuan, "An alternative view: When does SGD escape local minima?" in *Proc. Int. Conf. Machine Learning*, 2018, pp. 2698–2707.

[27] R. Kuditipudi, X. Wang, H. Lee, Y. Zhang, Z. Li, W. Hu, R. Ge, and S. Arora, "Explaining landscape connectivity of low-cost solutions for multilayer nets," in *Proc. Advances Neural Information Processing Systems*, 2019, pp. 14,574–14,583.

[28] J. D. Lee, M. Simchowitz, M. I. Jordan, and B. Recht, "Gradient descent only converges to minimizers," in *Proc. Conf. Learning Theory*, 2016, pp. 1246–1257.

[29] D. Li, T. Ding, and R. Sun, On the benefit of width for neural networks: Disappearance of bad basins. 2018. [Online]. Available: arXiv:1812.11039

[30] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, "Visualizing the loss landscape of neural nets," in *Proc. Advances Neural Information Processing Systems*, 2018, pp. 6391–6401.

[31] Y. Li and Y. Liang, "Learning overparameterized neural networks via stochastic gradient descent on structured data," in *Proc. Advances Neural Information Processing Systems*, 2018, pp. 6391–6401.

[32] S. Liang, R. Sun, J. D. Lee, and R. Srikant, "Adding one neuron can eliminate all bad local minima," in *Proc. Advances Neural Information Processing Systems*, 2018, pp. 4355–4365.

[33] S. Liang, R. Sun, Y. Li, and R. Srikant, "Understanding the loss surface of neural networks for binary classification," in *Proc. Int. Conf. Machine Learning*, 2018, pp. 2835–2843.

[34] S. Liang, R. Sun, and R. Srikant, Revisiting landscape analysis in deep neural networks: Eliminating decreasing paths to infinity. 2019. [Online]. Available: arXiv:1912.13472

[35] H. Lu and K. Kawaguchi, Depth creates no bad local minima. 2017. [Online]. Available: arXiv:1702.08580

[36] Y. Nesterov and V. Spokoiny, "Random gradient-free minimization of convex functions," *Found. Comput. Math.*, vol. 17, no. 2, pp. 527–566, 2017. doi: 10.1007/s10208-015-9296-2.

[37] Q. Nguyen, "On connected sublevel sets in deep learning," in *Proc. Int. Conf. Machine Learning*, 2019, pp. 4790–4799.

[38] Q. Nguyen and M. Hein, "The loss surface of deep and wide neural networks," in *Proc. 34th Int. Conf. Machine Learning*, 2017, pp. 2603–2612. doi: 10.5555/3305890.3305950.

[39] Q. Nguyen, M. C. Mukkamala, and M. Hein, "On the loss landscape of a class of deep neural networks with no bad local valleys," in *Proc. Int. Conf. Learning Representations*, 2019.

[40] I. Safran and O. Shamir, "Spurious local minima are common in two-layer relu neural networks," in *Proc. Int. Conf. Machine Learning*, 2018, pp. 4433–4441.

[41] R.-Y. Sun, "Optimization for deep learning: An overview," *J. Oper. Res. Soc. China*, vol. 8, pp. 249–294, June 2020. doi: 10.1007/s40305-020-00309-6.

[42] L. Venturi, A. Bandeira, and J. Bruna, Neural networks with finite intrinsic dimension have no spurious valleys. 2018. [Online]. Available: arXiv:1802.06384

[43] X.-H. Yu and G.-A. Chen, "On the local minima free condition of backpropagation learning," *IEEE Trans. Neural Netw.*, vol. 6, no. 5, pp. 1300–1303, 1995.

[44] C. Yun, S. Sra, and A. Jadbabaie, "Global optimality conditions for deep neural networks," in *Proc. Int. Conf. Learning Representations*, 2018.

[45] Y. Zhou and Y. Liang, "Critical points of neural networks: Analytical forms and landscape properties," in *Proc. Int. Conf. Learning Representations*, 2018.

[46] F. Zou, L. Shen, Z. Jie, J. Sun, and W. Liu, Weighted AdaGrad with unified momentum. 2018. [Online]. Available: arXiv:1808.03408

[47] M. Minsky and S. Papert, "Perceptrons," in *Neurocomputing: Foundations of Research*, J. A. Anderson and E. Rosenfeld, Eds. Cambridge, MA: MIT Press, pp. 1988, 157–169.

[48] B. Neyshabur and Z. Li, Towards understanding the role of over-parametrization in generalization of neural networks. 2019. [Online]. Available: arXiv:1805.12076

[49] Y. Li, T. Ma, and H. Zhang, "Algorithmic regularization in over-parameterized matrix sensing and neural networks with quadratic activations," in *Proc. 31st Conf. Learning Theory*, 2018, pp. 2–47.

[50] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, P. Tak, and P. Tang, On large-batch training for deep learning: Generalization gap and sharp minima. 2016. [Online]. Available: arXiv:1609.04836

[51] F. Pittorino et al., Entropic gradient descent algorithms and wide flat minima. 2020. [Online]. Available: arXiv:2006.07897

[52] C. Baldassi, F. Pittorino, and R. Zecchina, "Shaping the learning landscape in neural networks around wide flat minima," *Proc. Nat. Acad. Sci.*, vol. 117, no. 1, pp. 161–170, 2020. doi: 10.1073/pnas.1908636117.