Do Neural Networks Really Find the Best Parameters? Understanding the Global Landscape of Neural Networks

Bernardo Bianco Prado

Abstract

Neural networks are omnipresent in contemporary machine learning and are promising in dealing with the immense amount of data that exists in today's world. Because networks are so useful, it's essential to understand their behavior and their outputs. Especially because their outputs influence people's lives and experiences. This paper surveys a series of results that seek to understand when neural networks converge to a good output.

1 Introduction: Formulating Our Discussion Mathematically

Neural networks have become one of the most promising tools in contemporary machine learning problems. Unfortunately, these networks are as mysterious as they are useful. Because of their complexity (they are built from many function compositions), it's hard to predict what kind of parameter fit (good or bad) will be returned by the neural network. Exploring what can be said about these outputs of neural networks has been subject of great research in recent years and many partial discoveries have been publicized. Unfortunately, a definitive result is still lacking. In this article we review the exposition found in [1] which surveys results that seek to understand how good are the outputs of neural networks. All of the facts presented in this article are taken from [1] unless specified otherwise.

Let us first formulate our discussion in more mathematical terms. We are interested in whether the output of a neural network is good, and the measurement of "goodness" is usually performed in terms of some loss function $F(\boldsymbol{\theta})$ which is minimized by the best parameter $\boldsymbol{\theta}$. In other words, we are searching for

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^D} F(\boldsymbol{\theta})\,.$$

The most common methods of finding the optimal $\boldsymbol{\theta}^*$ are essentially improved variants of gradient descent. The issues arise because, since $F(\boldsymbol{\theta})$ is not convex, the gradient descent method can (in theory) get stuck in local minima or saddle points of the function $F(\boldsymbol{\theta})$. The article [1] surveys different results that try to understand the output of neural networks from two different angles. First, it seeks to understand whether it's possible that all local minima of $F(\boldsymbol{\theta})$ are in fact global minima. From a different angle, the article tries to understand whether the gradient descent method can avoid the sub-optimal local minima of $F(\boldsymbol{\theta})$ even if these bad local minima are present in the loss function.

1.1 Specifying the Objects of Study

For this discussion, we will need to talk about neural networks rigorously so it will be helpful introduce them in a mathematical language. Given an collection of input data $\boldsymbol{x}_1, ..., \boldsymbol{x}_n \in \mathbb{R}^{d_x}$ and output data $\boldsymbol{y}_1, ..., \boldsymbol{y}_n \in \mathbb{R}^{d_y}$, a fully connected neural network is a function $f_{\boldsymbol{\theta}} : \mathbb{R}^{d_x} \to \mathbb{R}^{d_y}$ given by

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{W}_L\phi(\boldsymbol{W}_{L-1}...\phi(\boldsymbol{W}_2\phi(\boldsymbol{W}_1\boldsymbol{x} + b_1) + b_2)... + b_{L-1})$$

where $\phi : \mathbb{R} \to \mathbb{R}$ is called an activation function that is applied to vectors in an element-wise fashion, each $\boldsymbol{W}_i \in \mathbb{R}^{d_i \times d_{i-1}}$ is a matrix and $\boldsymbol{\theta} = (\boldsymbol{W}_1, b_1, ..., \boldsymbol{W}_{L-1}, b_{L-1}, b_L)$. By definition, we have specified $d_0 = d_x, d_L = d_y$. Our hope is that the neural network $f_{\boldsymbol{\theta}}$ will predict the output $\boldsymbol{y}$ from the input $\boldsymbol{x}$ based on the available data inputs $\{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^n$. To ensure that the neural network manages succeed in this prediction, we are trying to find the best parameters $\boldsymbol{\theta}$ that will fit the input data. Mathematically, this is fomulated as a minimization problem

$$\min_{\boldsymbol{\theta}} F(\boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \ell(\boldsymbol{y}_i, f_\theta(\boldsymbol{x}_i))$$

where $\ell(\boldsymbol{y}, \boldsymbol{z})$ is a function that measures the "distance" between $\boldsymbol{y}$ and $\boldsymbol{z}$. Some examples are the Euclidean distance $\ell(\boldsymbol{y}, \boldsymbol{z}) = ||\boldsymbol{y} - \boldsymbol{z}||_2^2 = \sum_{j=1}^{d_y} (y_j - z_j)^2$ for regression problems and $\ell(\boldsymbol{y}, \boldsymbol{z}) = \log(1 + \exp(-\boldsymbol{y}^T \boldsymbol{z}))$ for binary classification problems.

2 A Promising Example: Linear Neural Networks

Let us start by discussing a simpler problem in which the map ϕ is just the identity map and so

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{W}_L(\boldsymbol{W}_{L-1}...(\boldsymbol{W}_2(\boldsymbol{W}_1\boldsymbol{x} + b_1) + b_2)... + b_{L-1})\,.$$

For simplicity, let us also assume $b_1, ..., b_{L-1} = 0$ so that

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{W}_L\boldsymbol{W}_{L-1}...\boldsymbol{W}_2\boldsymbol{W}_1\boldsymbol{x}\,.$$

We are also interested in the regression problem where $\ell(\boldsymbol{y}, \boldsymbol{z}) = ||\boldsymbol{y} - \boldsymbol{z}||_2^2$. The linear neural network minimization problem then becomes

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} ||\boldsymbol{y}_i - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)||_2^2 = \min_{\boldsymbol{\theta}} ||\boldsymbol{Y} - \boldsymbol{W}_L\boldsymbol{W}_{L-1}...\boldsymbol{W}_2\boldsymbol{W}_1\boldsymbol{X}||_F^2 \tag{1}$$

where $\boldsymbol{Y} = [\boldsymbol{y}_1, ..., \boldsymbol{y}_n]$, $\boldsymbol{X} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_n]$ are concatenation matrices of the data and $||\cdot||_F$ is the Frobenius norm.

The simplest example of this problem would be

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

One can see that this has a collection of global minima when $uv = 1$, these are the only local minima of this function, but it does have other critical points such as a saddle point at $(u, v) = (0, 0)$. This is an example of a network for which every local minimum is a global minimum. More generally, we have the following result.

Theorem 2.1 *If X, Y in the minimization problem (1) have full rank then every local minimum of the objective function is a global minimum.* □

In the example above, we have $X = 1$ and $Y = 1$ which are nonzero scalars and thus have full rank. This theorem generalizes the example for higher dimensional problems. Note that the result does not characterize critical points. For that we have a different theorem, which we write informally to get the intuition across

Theorem 2.2 (Characterization of Critical Points for Overparametrized Linear Networks) *Assume $n \geq d_x, d_y$ (there are more data points than the dimension of the input and output), $\boldsymbol{X}\boldsymbol{X}^T$ and $\boldsymbol{X}\boldsymbol{Y}^T$ are full rank, and $\boldsymbol{Y}\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{X}^T)^{-1}\boldsymbol{X}$ has distinct singular values. Assume further that the output and input layer have the lowest dimension out of all the layers of the network. Then for the problem (1), all critical points with $\boldsymbol{\theta} = (\boldsymbol{W}_1, ..., \boldsymbol{W}_L)$ so that $\boldsymbol{W}_L...\boldsymbol{W}_2\boldsymbol{W}_1$ being full rank is a global minimum and every critical point with $\boldsymbol{W}_L...\boldsymbol{W}_2\boldsymbol{W}_1$ being singular is a saddle point.* □

Observe that this theorem agrees with the example above. When $uv = 1$, then 1 is full rank and we have a global minimum. On the other hand, the critical point $u = v = 0$ has $uv = 0$ being singular and corresponds to a saddle point. This allowed us to completely classify critical points for certain overparametrized linear neural networks. Now we look at more general (nonlinear) overparametrized networks.

3 Moving Onward: Overparametrized Nonlinear Networks

As in the previous section, we start with a simple, one dimensional example of a nonlinear network. Consider

$$\min_{u,v\in\mathbb{R}} (u\phi(v) - 1)^2 \tag{2}$$

Consider, for example, when the activation function is the ReLU function $\phi(t) = \max\{0, t\}$. This has a sub-optimal local minimum at $u = 0, v = -1$! This is explained in the following theorem.

Theorem 3.1 *Let x, y be nonzero real numbers and consider the minimization problem*

$$\min_{u,v\in\mathbb{R}} (y - u\phi(vx))^2 .$$

Then the following are equivalent:

(I) The problem has no sub-optimal local minima.

(II) If $\phi(t) = 0$ then t is not a local maximum or minimum of ϕ. □

This is a unsatisfying result since ReLU is a popular choice of activation function. There is in fact a result that, generally, for sigmoid activation functions, there is no guarantee that there will not be sub-optimal local minima for the single layer, one dimensional network minimization problem. This is formalized below.

Proposition 3.2 *For $n \geq 3$ with the data points $x_1, ..., x_n \in \mathbb{R}$ all distinct and for an sigmoid activation function ϕ, there exist output data $y_1, ..., y_n \in \mathbb{R}$ so that the minimization problem*

$$\min_{u,v\in\mathbb{R}} \sum_{i=1}^{n} (y_i - u\phi(vx))^2$$

has a sub-optimal local minimum. □

In other words, for such sigmoid functions it's always possible that real world data will yield a neural network problem that has sub-optimal local minima. This proposition brings us to a halt in the discussion of sub-optimal local minima. We must then move away from this discussion of suboptimal local minima and towards something else. One useful notion of study is that of valleys and basins.

3.1 A Less Local Approach: Valleys and Basins

The idea behind this discussion is to look at the landscape of the objective function and move away from the pointwise perspective. We are interested in what the neighborhood of local minima looks like. A useful notion is that of a *spurious valley*. A spurious valley is a connected component of the sub-level set

$$\{\boldsymbol{\theta} : F(\boldsymbol{\theta}) \leq c\}$$

that contains no strict global minimum of the objective function.

If a function has no spurious valley, even if a sub-optimal local minimum exists on a sufficiently small sub-level set of the objective function, there is a non-increasing path from that local minimum to the global minimum. This is useful because it might prevent the gradient descent method from reaching and getting stuck at sub-optimal local minima. Therefore, we are interested in cases when there are no spurious valleys. To this end, we have the following theorem.

Theorem 3.3 *Suppose that a one dimensional, arbitrarily deep neural network $f_{\boldsymbol{\theta}}$ satisfies the following:*

1. *The activation function ϕ is strictly monotonic and $\phi(\mathbb{R}) = \mathbb{R}$.*
2. *The activation function is linearly independent from any shifted versions of itself. That is, for any $m \geq 2$ and all combination of coefficients $a_1, ..., a_m$ and $c_0, c_1, ..., c_m$ with the c_i distinct, if $c_0\phi(x) + c_1\phi(x - a_1) + ... + c_m\phi(x - a_m) = 0$, then we must have that $c_0 = c_1 = ... = c_m = 0$.*
3. *$d_{L-1} \geq d$.*
4. *All the training data points are distinct.*

Then the empirical loss objective function has no spurious valleys.

This would cover, for example $\phi(t) = t$ but not the ReLU activation function. In any case, this is a promising result and moving away from the local perspective of local minima might be useful. Following this philosophy, it is of interest to talk about set-wise local minima. In particularly, we are interested in *set-wise strict local minima*. For a function $f : \mathbb{R}^d \to \mathbb{R}^l$, the set S is a strict local minimum of f in the sense of sets if there is some $\epsilon > 0$ so that for all $\boldsymbol{x} \in S$ and all $\boldsymbol{y} \in \mathbb{R}^d \setminus S$ satisfying $||\boldsymbol{x} - \boldsymbol{y}||_2 \leq \epsilon$, we have $f(\boldsymbol{x}) < f(\boldsymbol{y})$. In other words, a set-wise local minimum of f is a set so that the value of f all its points close to its boundary are smaller than all the nearby points outside of its boundary.

The notion of a set-wise strict local minimum gives rise to the definition of a *sub-optimal basin*, which is a set-wise local minimum that contains no global minima. This notion is essentially the set version of a sub-optimal local minimum. There is a result analogous to the theorem above to sub-optimal basins.

Theorem 3.4 *Suppose that a one dimensional, arbitrarily deep neural network $f_{\boldsymbol{\theta}}$ satisfies the following:*

1. *There is a k so that the k-th entry of all the input data $(\boldsymbol{x}_i)_k$ are distinct.*
2. *$d_{L-1} \geq d$.*
3. *The activation function ϕ is continuous.*

Suppose, in addition that the loss function $\ell(\boldsymbol{u}, \boldsymbol{v})$ is convex with respect to $\boldsymbol{v}$. Then the empirical loss objective function has no sub-optimal basins.

These results show a promising direction in understanding the behavior of descent methods for neural networks. These are the main theoretical results covered in the paper regarding the geometric landscape of neural networks. We now discuss some attempts at eliminating sub-optimal local minima from the network minimization problem.

4 Getting Rid of the Issue: Removing Bad Local Minima from Nonlinear Neural Networks

We are interested in identifying the regions where the there are no sub-optimal local minima for the objective function of the neural network problem. For this, we are mostly interested in the last layer of the neural networks. In particular, we rewrite the neural network as

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{W}_L \boldsymbol{Z}_{\boldsymbol{\theta}}(\boldsymbol{x}) \,.$$

Where $\boldsymbol{Z}_{\boldsymbol{\theta}}$ essentially encompasses all the other layers of the neural network. Rewriting the problem like this is useful because we can impose conditions on $\boldsymbol{Z}_{\boldsymbol{\theta}}$ so that the minimization problem has no bad local minima as long as $\boldsymbol{x}$ is restricted to a subset of $\mathbb{R}^{d_x}$. An example presented below.

Claim 4.1 *If $\boldsymbol{\theta}^*$ is a local minimum of $F(\boldsymbol{\theta}^*) = \sum_{i=1}^n ||\boldsymbol{y}_i - f_{\boldsymbol{\theta}^*}(\boldsymbol{x}_i)||_2^2$ so that the matrix*

$$\boldsymbol{Z}_{\boldsymbol{\theta}^*}(\boldsymbol{X}) = (\boldsymbol{Z}_{\boldsymbol{\theta}^*}(\boldsymbol{x}_1), ...\boldsymbol{Z}_{\boldsymbol{\theta}^*}(\boldsymbol{x}_1))$$

is full rank, then $\boldsymbol{\theta}^*$ is a global minimum.

Further, if the derivatives of the activation function $\phi^{(k)}(0)$ for $k = 1, ..., n-1$ are nonzero, then the set where $\{\boldsymbol{\theta} : \boldsymbol{Z}_{\boldsymbol{\theta}}(\boldsymbol{X})$ has full rank$\}$ is dense. □

This result allows us to avoid local sub-optimal local minima with almost sure probability whenever the activation function satisfies the derivative condition in the claim. Unfortunately, this does not cover, for example, non-smooth activation functions such as ReLU.

An alternative is to modify the network and the objective function for the minimization problem so that the problem yields better results. Suppose we modify the neural networks by creating a new function $\tilde{f}$ so that

$$\tilde{f}_{\tilde{\boldsymbol{\theta}}}(\boldsymbol{x}) = f_{\boldsymbol{\theta}}(\boldsymbol{x}) + a\exp(\boldsymbol{w}^T\boldsymbol{x} + b)$$

where $\tilde{\boldsymbol{\theta}} = (\boldsymbol{\theta}, a, \boldsymbol{w}, b)$. Suppose that the loss function ℓ is the logistic loss function

$$\ell(\boldsymbol{y}, \boldsymbol{z}) = \log(1 + \exp(-\boldsymbol{y}^T\boldsymbol{z}))$$

then we consider the minimization problem

$$\min_{\tilde{\boldsymbol{\theta}}} \tilde{F}(\tilde{\boldsymbol{\theta}}) = \min_{\tilde{\boldsymbol{\theta}}} \sum_{i=1}^n \ell(\boldsymbol{y}_i, \tilde{f}_{\tilde{\boldsymbol{\theta}}}(\boldsymbol{x}_i)) + \frac{\lambda a^2}{2}\,.$$

Then we have the following result.

Theorem 4.2 *For the minimization problem above, we have*

1. *$\tilde{F}(\tilde{\boldsymbol{\theta}})$ has at least one local minimum.*
2. *At every local minimum, $a = 0$.*
3. *If $\tilde{\boldsymbol{\theta}}^* = (\boldsymbol{\theta}^*, a^*, \boldsymbol{w}^*, b^*)$ is a local minimum of $\tilde{F}(\tilde{\boldsymbol{\theta}})$, then it is also a global minimum of $\tilde{F}(\tilde{\boldsymbol{\theta}})$ and $\boldsymbol{\theta}^*$ is a global minimum of*

$$F(\boldsymbol{\theta}) = \sum_{i=1}^n \ell(\boldsymbol{Y}_i, \tilde{f}_{\tilde{\boldsymbol{\theta}}}(\boldsymbol{x}_i))\,.$$

□

This result seems like it solves all of our problems! At least for the logistic regression case. There is one issue, however, and that is that even though there are no sub-optimal local minima of this new minimzation problem, there may be a decreasing path of $\tilde{\boldsymbol{\theta}}(s)$ for $s > 0$ so that $||\tilde{\boldsymbol{\theta}}(s)||_2 \to \infty$ as $s \to \infty$ and $\tilde{F}(\tilde{\boldsymbol{\theta}}(s))$ is strictly decreasing. This means that gradient descent may never converge to a local minimum and instead it may just send the $\tilde{\boldsymbol{\theta}}$ to infinity.

There is a way to avoid that for certain neural networks. For example, consider the one layer, one dimensional ReQU neural network defined by

$$f_{\boldsymbol{\theta}}(x) = \sum_{j=1}^m a_i(\max\{\boldsymbol{w}_j x + b_j\})^2 = \sum_{j=1}^m a_j\big(\text{ReQU}(\boldsymbol{w}_j x + b_j)\big)^2\,,$$

where $m \geq n+1$ with n being the number of input data points. As above, suppose the loss function is logistic regression and suppose we are minimizing the objective function

$$F(\boldsymbol{\theta}) = \sum_{i=1}^{n} \ell(y_i, f_{\boldsymbol{\theta}}(x_i)) + \frac{1}{3}\sum_{j=1}^{m} \lambda_j \Big(|a_j|^3 + 2(||\boldsymbol{w}_j||_2^2 + b_j^2)^{3/2} \Big)$$

where $\boldsymbol{\theta} = (a_1, \boldsymbol{w}_1, b_1, ..., a_m, \boldsymbol{w}_m, b_m)$, and where $\boldsymbol{\lambda} = (\lambda_1, ..., \lambda_m)$ is a parameter vector that needs to be tuned in the problem. For the right $\boldsymbol{\lambda}$, one can prove that every local min achieves zero training error and thus every local minimum would be a global minimum. In particular, we have the following

Theorem 4.3 *If $m \geq n+1$ and $F(\boldsymbol{\theta})$ is defined as above, then for each collection of input and output data $x_1, ..., x_n, y_1, ..., y_n \in \mathbb{R}$, there is a $\lambda_0 > 0$ and a set $\mathscr{C} \subset \mathbb{R}$ of measure zero so that for any $\boldsymbol{\lambda} \in (0, \lambda_0)^m \setminus \mathscr{C}$, we have*

1. *The objective function $F(\boldsymbol{\theta})$ is coersive, that is*

$$\lim_{||\boldsymbol{\theta}|| \to \infty} F(\boldsymbol{\theta}) = +\infty \, .$$

2. *Every local minimum $\boldsymbol{\theta}^*$ of the function $F(\boldsymbol{\theta})$ is a global minimum of $F(\boldsymbol{\theta})$ and achieves zero training error.* □

This says that basically whatever $\boldsymbol{\lambda}$ one chooses, as long as its entries are small enough, gradient descent will likely always converge to a global minimum of $F(\boldsymbol{\theta})$.

We have covered useful properties of the minimization problem that will guarantee a desirable landscape of the objective function for the minimization problem. We have not, however discussed whether those properties translate to convergence of gradient descent to global minima. This is what we focus on below.

5 Was It All Worth It? The Behavior of Gradient Descent on the Discussed Problems

The hope is that for objective functions with the properties discussed above then for most initializations, gradient descent will converge to a global minimum. For example, if a function has sub-optimal basins the hope is that the sub-optimal basins have small measure and thus, gradient descent will converge to a global minimum with high probability. The goal result is outlined below.

Principle (Blueprint for Gradient Descent) *If F is an objective function represents the neural network minimization problem. Let $\boldsymbol{\theta}^*$ be the point of convergence of gradient descent or stochastic gradient descent and F^* be the minimum value of $F(\boldsymbol{\theta})$. Then there are small constants δ, ϵ so that*

$$\mathbb{P}(F(\boldsymbol{\theta}^*) < F^* + \epsilon) > 1 - \delta \, .$$

Intuitively, this means that the gradient descent method will converge to a almost optimal value for F with high probability. Unfortunately, there is still not a lot of results that confirm this principle theoretically. Because we the conjecture deals with probability measures, stochastic gradient descent has proved more fitting to this discussion as it favors a probabilistic perspective.

There are some results, however, that guide us in the direction of this principle. The two strategies to achieve this are first showing that for almost all initial conditions avoid the bad regions of the objective function landscape. The other strategy is to show that whenever gradient descent reaches a bad region, it will likely leave it and move on towards a global minimum.

Some results are available for ultra-wide neural networks. For an example, consider the matrix $\boldsymbol{G}(\boldsymbol{\theta}) = (\text{grad}_{\boldsymbol{\theta}}\, f_{\boldsymbol{\theta}}(\boldsymbol{x}_1), ..., \text{grad}_{\boldsymbol{\theta}}\, f_{\boldsymbol{\theta}}(\boldsymbol{x}_n))$ and $\boldsymbol{K}(\boldsymbol{\theta}) = \boldsymbol{G}(\boldsymbol{\theta})^T\boldsymbol{G}(\boldsymbol{\theta})$ and suppose $d_y = 1$. Suppose the loss function ℓ is the Euclidean distance squared. Then we have the following result.

Claim 5.1 *We have the following:*

1. *If θ^* is a critical point of the objective function $F(\boldsymbol{\theta})$ and $\boldsymbol{K}(\boldsymbol{\theta}^*)$ has full rank, then $\boldsymbol{\theta}^*$ is a global minimum and $F(\boldsymbol{\theta}^*) = 0$.*

2. *If $\boldsymbol{\theta}_k$ for $k = 0, 1, ...$ are the iterates of gradient descent and if there is some constant $c > 0$ so that*

$$\lambda_{min}(\boldsymbol{G}(\boldsymbol{\theta}_k)) \geq c$$

for all $k = 0, 1, ...$, then we have that and limit point $\boldsymbol{\theta}^*$ of the sequence $\{\boldsymbol{\theta}_k\}_{k=0}^{\infty}$ which is a critical point of $F(\boldsymbol{\theta})$ is a global minimum of $F(\boldsymbol{\theta})$ with $F(\boldsymbol{\theta}^*) = 0$.

□

These results correspond to ultra wide neural networks and are very strong. The unfortunate fact is that many neural networks of interest are not ultra-wide and thus this does not necessarily apply to many cases of interest. This summarizes some results for gradient descent behaving well with certain neural network problems.

6 Discussion and Conclusion

This paper covers a comprehensive collection of results regarding neural networks. One question I had was whether some of the nonsmoothness of ReLU can be fixed by looking at the function

$$r(t) = \begin{cases} 0 & \text{if } t \leq 0 \\ (1 + t^2)^{1/2} - 1 & \text{if } t > 0 \end{cases}$$

which is a commonly used function in mathematical analysis that asymptotically behaves like $|t|$ as $t \to \infty$ and is smooth at $t = 0$.

In the future, I hope to implement neural networks and observe the behavior of gradient descent with different activation functions. I also hope to read some of the proofs for the results more deeply to understand ideas behind the behavior of neural networks. I have learned a lot from this paper and now I have a deeper knowledge of how neural networks behave and I understand a bit more about the theory of these essential objects in contemporary machine learning. I will surely use some of these results to better choose the setup whenever I'm using neural networks in the future.

References

[1] Ruoyu Sun, Dawei Li, Shiyu Liang, Tian Ding and R Srikant. *The Global Landscape of Neural Networks: An Overview*. 2020. https://arxiv.org/abs/2007.01429