The Benign Loss Landscape of Neural Networks

Shuai Li
lishuai918@gmail.com

August 16, 2019

Abstract

We study the loss landscape of NNs, and prove under a set of conditions, for *large-size nonlinear deep* NNs with a class of loss functions $\mathcal{L}_0$, including the hinge loss, *all local minima are global minima with zero loss errors*. For the class $\mathcal{L}_0$ of loss functions, the proof enables a complete characterization of the stationary points of the NNs' loss landscape: eigenvalues of the Hessian of the NNs are distributed *symmetrically w.r.t. zero*, and concentrate increasingly towards zero as the risk decreases. We also show the loss landscape of a practical NN throughout training conforms to the one characterized.

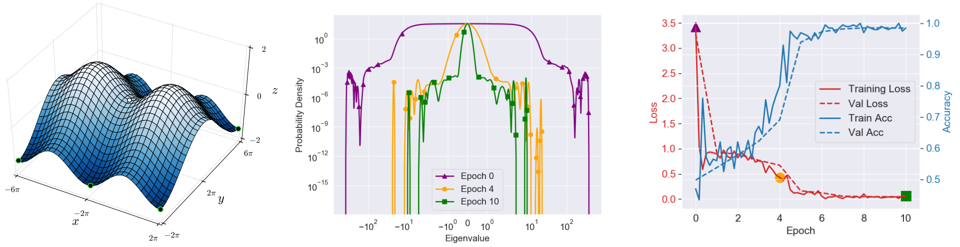
1 Introduction

Deep neural networks (NNs) are biologically inspired families of algorithms that model the perception part of intelligence, e.g., image recognition, in an astoundingly successful way, e.g., empowering human-level image classification system [19]. It is tempting to hypothesize that it may hold the key to understanding at least the perception intelligence [26]. However, the progress in the field so far has been largely driven by application-oriented interests, and a mathematical characterization has lagged behind. Among the theoretical aspects of NNs, we focus on the optimization behaviors in this work.

The optimization problem of NNs has been a long standing issue for decades that attracts many attempts to understand it. Numerical positive works focus on shallow NNs [2; 44; 4; 31; 53; 46; 17; 48; 50; 47; 5; 9; 14]; deep linear NNs or deep NNs with linear algebraic based conditions [36; 37; 43; 18; 29; 32; 52; 38; 45; 10]; deep nonlinear NNs with unrealistic mean-field type assumptions [7; 6; 40], i.e., neuron activation is independent with input data, or weights are identically independent distributed; and deep nonlinear networks with literally infinite layer width [23]. A detailed related works section is present in appendix B. In this paper, we present results that characterize the optimization behaviors of deep nonlinear neural networks that are observed in NNs used in practice.

As well known, given a risk function, the landscape of the function is characterized by the *eigen-spectrum (the distribution of eigenvalues)* of its Hessian: when all eigenvalues are positive, we are at a local minimum (black dots in fig. 1a); when all eigenvalues are negative, we are at a local maximum; otherwise, *we are at a saddle point, implying we have extra directions to further minimize the loss (the white dot in fig. 1a)*. Under a set of assumptions 3.2 3.3 A.2 3.5 A.3, we show in theorem 4.1 that the loss landscape of the nonconvex optimization problem of NNs with a class of losses (which includes *hinge loss*) is rather benign; that is, *all local minima are global minima*. We note that though widely used in practice, the class of functions is relatively simple, and does not include quadratic loss, or cross entropy loss. More specifically, the theorem shows that the landscape of NNs with loss functions in a class $\mathcal{L}_0$ (specified in definition 1 later) satisfies the following characterization:

a) all local minima are global minima with zero errors.
b) all non-zero-error stationary points are saddle points with half of the non-zero eigenvalues negative.
c) The regions around the minima are flat basins, and the eigen-spectrum of the Hessian of the risk function is increasingly concentrated around zero as the risk value decreases.



(a) (b) (c)

Figure 1: (a) Illustration of a loss landscape with saddle points; the white dot at the middle is a saddle points. (b) The numerically computed eigen-spectrum of Hessian of LeNet (cross entropy loss replaced with hinge loss) being trained on a modified MNIST dataset for binary classification at epoch 0, 4, 10. *Note that the axes are in logarithmic scale.* The training losses at epoch 0, 4, 10 are also marked on the loss curve in (c). The eigen-spectrum is computed with the Lanczos spectrum approximation algorithm [33; 39; 13] (details in appendix D). (c) The loss and accuracy curve of our experiment.

Furthermore, we find that the characterization of loss landscape exists in practical NNs. We train a LeNet [30] (cross entropy loss replaced with hinge loss) on a MNIST dataset modified for binary classification, shown in fig. 1. The experiment details can be found in appendix D. The theorem states that the eigen-spectrum of the Hessian is symmetric around zero; that is, half of the non-zero eigenvalues of the Hessian are negative. The symmetry is shown in fig. 1b. Thus, every non-zero loss stationary point in the loss landscape can be further minimized. Yet, if all stationary points are saddle points, how can local minima be found? Characterization c) tells us the eigen-spectrum of the Hessian would increasingly concentrate on zero, and the loss landscape would become increasingly flat, as shown in fig. 1b, where the eigen-spectrums of the NN's Hessian at epoch 0, 4, 10 during training concentrate increasingly around zero in an exponential way. The minima are achieved when all samples $\boldsymbol{x}$ are classified rightly — the risk is zero; in this case, the Hessian becomes a zero matrix, and all eigenvalues are zero. The training loss and accuracy curves are shown in fig. 1c.

The theorem *implies* that if asp. 3.2 3.3 A.2 3.5 A.3 are made satisfied throughout training, we can always reach the zero training error for binary classification problems with NNs, not only just for easy tasks like MNIST. The key ideas, i.e., measure transport and concentration of measure, in principle can be generalized to any loss functions, and building on these ideas, we believe extending the results on function class $\mathcal{L}_0$ to any convex functions is possible. It is a *blessing* of dimensionality that makes non-convex optimization tractable, even elegant and beautiful in our opinion. However, though we would remark in section 3.5 that initialization [15] and normalization [22] techniques widely used that make NNs optimizable probably make NNs satisfy our assumptions, there are still gap between our results and techniques that could provable optimize NNs *as we wish*: our results only apply for relatively simple loss functions, e.g., hinge loss; training might be trapped in saddle points without reaching minima; the result does not involve generalization (different local minima are of different generalization ability, leading to different test errors), which is a challenge as significant as optimization; etc. Despite those gaps, we believe a plausible perspective is provided by this work to theoretically understand in *what* context and *why* the optimization of NNs works. For instance, it would be imminent future works to explore techniques that would provably optimize NNs (not *as we wish* mentioned earlier, since there are still issues such as generalization) based on our assumptions, or explore theoretical justification for normalization techniques, e.g. Batch Normalization [22].

Outline. In the rest of the paper, we present the key ideas that lead to our results, i.e., theorem 4.1. We will present two key ideas. First, we show that NNs are hierarchical measure transport in section 2, and thus the Hessian of a NN is a symmetric random matrix (section 3.2). Second, we introduce a set of assumptions that characterize the weak correlation in the Hessian in section 3.4, and present the insight in section 4 that weak correlation in the Hessian of NNs induce the measure concentration phenomenon that leads to the benign loss landscape. The main result, i.e. theorem 4.1, is also present in section 4.1. A proof sketch that connects the ideas present is provided in section 4.3. The full proof is in appendix A.

Contribution.

- We prove under a set of conditions, for *large-size nonlinear deep* NNs with a class of loss functions $\mathcal{L}_0$, including the hinge loss, *all local minima are global minima with zero loss errors*.
- We provide a complete characterization of the stationary points of the NNs' loss landscape for the class $\mathcal{L}_0$ of loss functions: eigenvalues of the Hessian of the NNs are distributed *symmetrically w.r.t. zero*, and concentrate increasingly towards zero as the risk decreases.
- We corroborate the theoretical results with an experiment that train LeNet on MNIST, and show the loss landscape of the NN throughout training conforms to the one prescribed.

2 Neural networks as hierarchical measure transport

In this section, we bridge the gap between probability and optimization in NNs by formulating NNs as a hierarchical measure transport. It serves as the foundation of the landscape analysis carried on later.

Notations. All scalar functions are denoted as normal letters, e.g., f; bold, lowercase letters denote vectors, e.g., $\boldsymbol{x}$; bold, uppercase letters denote matrices, or random matrices, e.g., $\boldsymbol{W}$; normal, uppercase letters denotes random elements/variables, e.g., H. r.v. and r.v.s are short for random variable, random variables respectively. To index entries of a matrix, we denote $\mathbb{J} = [N] := \{1, \ldots, N\}$, the ordered pairs of indices by $\mathbb{I} = \mathbb{J} \times \mathbb{J}$. For $\alpha \in \mathbb{I}, A \subseteq \mathbb{I}, i \in \mathbb{J}$, given a matrix $\boldsymbol{W}$, $w_\alpha, \boldsymbol{W}_A, \boldsymbol{W}_{i:}, \boldsymbol{W}_{:i}$ denotes the entry at α, the vector consists of entries at A, the ith row, ith column of $\boldsymbol{W}$ respectively. Given two matrices $\boldsymbol{A}, \boldsymbol{B}$, the curly inequality $\preceq$ between matrices, i.e., $\boldsymbol{A} \preceq \boldsymbol{B}$, means $\boldsymbol{B} - \boldsymbol{A}$ is a positive definite matrix. Similar statements apply between a matrix and a vector, and a matrix and a scalar. $\succeq$ is defined similarly. $:=$ is the "define" symbol, where $x := y$ defines a new symbol x by equating it with y. tr denotes matrix trace. $\mathrm{dg}(\boldsymbol{h})$ denotes the diagonal matrix whose diagonal is the vector $\boldsymbol{h}$. $\kappa(\cdot, \cdot)$ denotes covariance, e.g., $\kappa(w_\alpha, w_\beta)$ as the covariance between $w_\alpha, w_\beta, \alpha, \beta \in \mathbb{I}$. or covariance between functions f, g of entries in $\boldsymbol{W}$, i.e., $\kappa(f(\boldsymbol{W}_A), \kappa(w_\alpha, w_\beta)$. $||\cdot||$ denotes 2-norm if not specified otherwise. $\Re$ and $\Im$ take the real and imaginary part of its operand (a complex number) respectively. All the remaining symbols are defined when needed, and should be self-clear in the context.

2.1 Measure transport

NNs are widely taken as universal function approximators, and since a probability distribution is a function, NNs can also approximate probabilities. However, underlying the widely perceived picture described, a richer structure exists: the probability distribution a NN approximates is the probability measure transport from the data to the hidden neurons. To describe the structure, we introduce measure transport. Given measurable spaces $(E, \mathcal{E})$, $(F, \mathcal{F})$, a measurable mapping $T : E \to F$, and

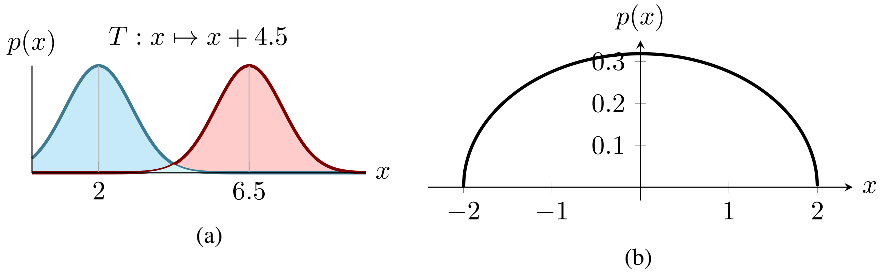


Figure 2: **(a)**: Measure Transport Illustration. The blue shaded area on the left is the probability measure of a random variable X taking value in $[-\infty, +\infty]$. It is a normal distribution centering at 2. A transport map $T : x \mapsto x + 4.5$ transports the measure of X to the measure of a new random variable $Y := X + 4.5$. The pushforward measure is the red shaded area on the right, centering at 6.5. That is, the measure transported from X to Y by T. **(b)**: Eigen-spectrum of Wigner matrix. The x-axis is the numerical value of eigenvalues, and the y-axis is the probability of obtaining a eigenvalue of a specific numeric value if we randomly sample a eigenvalue of all the eigenvalues of $\boldsymbol{A}_N$.

a measure $\mu : \mathcal{E} \rightarrow [0, +\infty]$, the *pushforward probability measure* of μ by T is defined to be the measure $\nu : \mathcal{F} \rightarrow [0, +\infty]$ given by

$$\nu(B) = \mu(T^{-1}(B)), B \in \mathcal{F}$$

The mapping T is often called *transport map*. Without further delving into the formalism of probability measure theory, given a random variable (r.v.) X with a probability measure μ, a measurable mapping T, and the r.v. $Y := T(X)$ with probability measure ν obtained by applying the measurable mapping T on X, we say the *measure transported* from X to Y is the pushforward probability measure ν of μ by T. **An example of measure transport is shown in fig. 2a**.

2.2 Statistical learning

With measure transport, we are ready to introduce the learning problem of supervised NNs.

Assume an instance space $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is the space of input data, and $\mathcal{Y}$ is the label space. $Z := (X, Y)$ are the r.v.s with an unknown distribution μ, from which we draw samples. We use $S_m = \{z_i = (\boldsymbol{x}_i, y_i)\}_{i=1}^m$ to denote the training set of size m whose samples are drawn independently and identically distributed (i.i.d.) by sampling Z. Given a loss function l, the goal of learning is to identify a function $T : \mathcal{X} \mapsto \mathcal{Y}$ in a hypothesis space (a class $\mathcal{T}$ of functions) that minimizes the expected risk

$$R(l \circ T) = \mathbb{E}_{Z \sim \mu} \left[l\left(T(X), Y\right) \right],$$

Since μ is unknown, the observable quantity serving as the proxy to the true risk $R(f)$ is the empirical risk

$$R_m(l \circ T) = \frac{1}{m} \sum_{i=1}^{m} l\left(T(\boldsymbol{x}_i), y_i\right) \tag{1}$$

Thus, given a mapping T, e.g., a NN, **the risk function R_m measures the empirical discrepancy between the probability measure $T(X)$ transported from X and the probability measure of Y.** Thus, the learning of NNs is to learn parametric approximation of conditional measure of Y through minimizing a surrogate risk.

2.3 Neural networks

Not only the ultimate output of a NN, $T(X)$, is a r.v., so are the activation in the intermediate layers of a NN. We characterize the finer stochastic structure in NNs using Multiple Layer Perceptron (MLP) in this section.

Denote g as the activation function, $\boldsymbol{W}$ the linear transform, X the input, $\hat{H}$ the output activation, of a layer of a NN. We have

$$\begin{aligned} \hat{H} &:= g(\boldsymbol{W}X) = \text{ReLU}(\boldsymbol{W}X) \\ \text{ReLU}(\boldsymbol{W}X) &:= \boldsymbol{W}X \odot H = \text{dg}(H)\boldsymbol{W}X \\ H &:= \mathbf{1}^{\boldsymbol{W}X}, \mathbf{1}_i^{\boldsymbol{W}X} = \begin{cases} 1, & \text{if } (\boldsymbol{W}X)_i > 0 \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{2}$$

where $\odot$ denotes Hadamard product (element-wise multiplication); ReLU denotes the Rectified Linear Unit; dg(H) is the diagonal matrix whose diagonal is H. We use the widely used activation function ReLU as an example. The derivation in this paper can be trivially generalized to activation functions like Swish [42], or variants of ReLU [19; 8] etc. Generalizing to classic activation functions like Sigmoid or Tanh may need some more involved matrix calculus, but is fundamentally similar.

$\hat{H}, H$ are both r.v.s created by transporting measure of X through map $g(\boldsymbol{W}X), \mathbf{1}^{\boldsymbol{W}X}$ respectively, thus they are *stochastic* (which is a tautology). Repeat the process, and **we can write a MLP with a single output neuron as the following transport map**

$$T(X; \boldsymbol{\theta}) = X^T \Pi_{i=1}^{L-1} \boldsymbol{W}_i \text{dg}(H_i) \boldsymbol{\alpha} \tag{3}$$

where $\boldsymbol{\alpha}$ is the weights of the last layer of a NN, and is a vector; i is the layer index; dg(H_i) is the diagonal matrix whose diagonal is H_i; $\boldsymbol{\theta}$ is the parameters of T, a.k.a. $\{\boldsymbol{W_i}\}_{i=1,\ldots,L-1}, \boldsymbol{\alpha}$; L is the layer number.

3 Assumptions on the Hessian of a restricted class of loss functions

In this section, after shortly presenting the settings of theorem 4.1, we set out to explain the assumptions of theorem 4.1. To begin with, we present an *intuitive* picture behind the assumptions here. It's widely conjectured that the patterns in NNs are coded hierarchically, where neurons in lower layers encode low-level patterns, e.g., textures, neurons in higher layers encode semantic patterns, e.g., objects, and the **activation paths** between neurons in the lower and higher layers encode the *composition relationship* between these patterns (the connection between composition relationship and activation paths is not established formally, but mere serves as an analog to help understanding). Overall, the assumptions characterize certain conditions on the strength and the strength of correlation between activation paths (**note that the word "correlation" used throughout this paper only refers to a qualitative relationship, not anything formally mathematic**). An illustration of activation paths is shown in fig. 3a. They could be heuristically understood as:

1. **The NN is of large enough size (Asp. 3.2).**
2. **The strength of each path, and the correlation between paths could be strong, but bounded, and comparable (Asp. 3.4 3.5).**
3. **A path does not correlate with the majority of the other paths (Asp. 3.6).**

Informally, the assumptions collectively state that NNs should be large to contain enough paths to encode hierarchical patterns; each pattern encoded should be treated roughly equally instead of being biased toward; many independent patterns are encoded (As an example, an object e.g., a person, contains a certain pattern, e.g, the leg of a person, does not need a pattern of another object, e.g., the leg of a chair. Those patterns only correlate at very low level, e.g., textures). Thus, to analogize it to bio-brains, NNs can reach zero errors when they are of large capacity and be open-minded to incorporate patterns at equal footing and tolerate mutually autonomous patterns.

3.1 The restricted loss function class $\mathcal{L}_0$

Our goal is to investigate the fundamental principle that makes $R_m(T)$ in eq. (1) tractable when T is a NN. To do this, we study the risk landscape by studying the Hessian of $R_m(T)$ of a particular class of loss functions. To motivate the class of losses, we calculate the Hessian of eq. (1) when T is given by eq. (3). The choice of a NN with a single output neuron simplifies the problem considerably: it reduces a weighted summation of a great number of matrices to a single matrix, though we do not want to digress too much to elaborate the simplification. The calculation gives

$$\frac{d^2}{d\boldsymbol{\theta}^2}l(T(\boldsymbol{x};\boldsymbol{\theta}),y) = l''(T(\boldsymbol{x};\boldsymbol{\theta}),y)\frac{d}{d\boldsymbol{\theta}}T(\boldsymbol{x};\boldsymbol{\theta})\frac{d}{d\boldsymbol{\theta}}T(\boldsymbol{x};\boldsymbol{\theta})^T + l'(T(\boldsymbol{x};\boldsymbol{\theta}),y)\frac{d^2}{d\boldsymbol{\theta}}T(\boldsymbol{x};\boldsymbol{\theta}) \tag{4}$$

Note that $\boldsymbol{x}, y$ are realizations sampled from X, Y now. To further simplify the problem, we restrict the class of loss functions to class $\mathcal{L}_0$.

Definition 1 (Function class $\mathcal{L}_0$)**.** *We study the class $\mathcal{L}_0$ of functions l and the class of NNs T, such that for $l \in \mathcal{L}_0$, it satisfies:*

1. *$l : \mathbb{R} \to \mathbb{R}^+$, when y is taken as a constant;*
2. *l is convex;*
3. *the second order derivatives $\frac{d^2}{d\boldsymbol{x}^2}l$ is zero;*
4. *$\min_x l(\boldsymbol{x}, y) = 0$, while for T it satisfies: $\dim(T(\boldsymbol{x};\boldsymbol{\theta})) = 1$.*

The restriction allows us to study the most critical aspect of the risk function of supervised NNs by making the first term on the right side of the equation in eq. (4) zero, while the second term a single matrix. The class of l includes important loss functions like the **hinge loss** $\max(0, 1 - \hat{H}_L Y)$, and the absolute loss $|\hat{H}_L - Y|$, which were first studied in Choromanska et al. [6].

3.2 Hessian of neural networks with loss in the function class $\mathcal{L}_0$

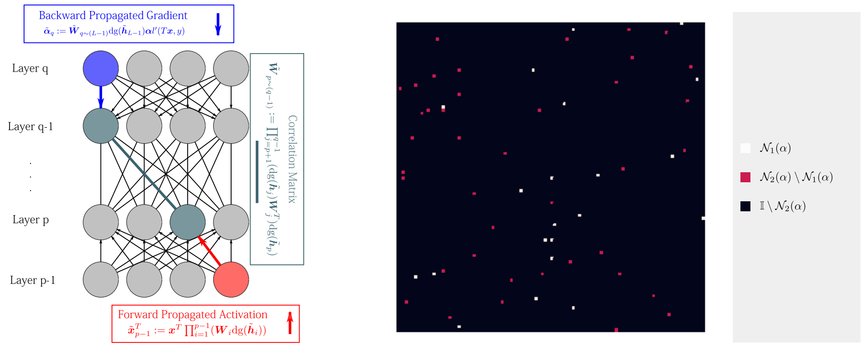
Denote $\boldsymbol{H}_{pq}$ as in eq. (5) (the equation should be read vertically). The operands of the Kronecker product are shortened as $\tilde{\boldsymbol{\alpha}}_q, \tilde{\boldsymbol{W}}_{p\sim(q-1)}, \tilde{\boldsymbol{x}}_{p-1}^T$ on the right most side in eq. (5), which corresponds to physical meaning that is explained in section 3.3 later and illustrated in fig. 3a. We have eq. (4), the Hessian of l, as in eq. (6). The calculation is deferred to appendix A.1.

$$\begin{aligned}\boldsymbol{H}_{pq} =& l'(T\boldsymbol{x}, y)\mathrm{dg}(\tilde{\boldsymbol{h}}_q)\Pi_{k=q+1}^{L-1}(\boldsymbol{W}_k\mathrm{dg}(\tilde{\boldsymbol{h}}_k))\boldsymbol{\alpha} && = \tilde{\boldsymbol{\alpha}}_q \\ & \otimes \Pi_{j=p+1}^{q-1}(\mathrm{dg}(\tilde{\boldsymbol{h}}_j)\boldsymbol{W}_j^T)\mathrm{dg}(\tilde{\boldsymbol{h}}_p) && \otimes \tilde{\boldsymbol{W}}_{p\sim(q-1)} \\ & \otimes \boldsymbol{x}^T\Pi_{i=1}^{p-1}(\boldsymbol{W}_i\mathrm{dg}(\tilde{\boldsymbol{h}}_i)) && \otimes \tilde{\boldsymbol{x}}_{p-1}^T\end{aligned} \tag{5}$$

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{H}_{12}^T & \dots & \boldsymbol{H}_{1L}^T \\ \boldsymbol{H}_{12} & \boldsymbol{0} & \dots & \boldsymbol{H}_{2L}^T \\ \vdots & \ddots & \ddots & \vdots \\ \boldsymbol{H}_{1L} & \boldsymbol{H}_{2L} & \dots & \boldsymbol{0} \end{bmatrix} \tag{6}$$

Intuitive understanding of eq. (5) is provided in section 3.3 shortly. For now, we just need to focus on the form of eq. (6), and ignore the complicated details, e.g., Kronecker products. Notice that $\boldsymbol{x}$ and $\tilde{\boldsymbol{h}}$ (subscripts omitted) are both realizations of r.v.s X and H. Thus, **Hessian $\boldsymbol{H}$ is a matrix obtained by applying a complicated transport map on X. The matrix is a symmetric random matrix,** the randomness of which comes from data. It implies the Hessian $\boldsymbol{H}$ of $R_m(T)$ is a random matrix created by summing random matrices, each of which is a gradient l' multiplies the Hessian of T at example z_i. **Its eigen-spectrum can be studied through random matrix theory** [49]. Thus, we can have a complete characterization of the landscape of $R_m(T)$ of eq. (1).

3.3 Entries of Hessian as activation paths


Backward Propagated Gradient
$\tilde{\boldsymbol{\alpha}}_q := \tilde{\boldsymbol{W}}_{q\sim(L-1)}\mathrm{dg}(\tilde{\boldsymbol{h}}_{L-1})\boldsymbol{\alpha}l'(T\boldsymbol{x}, y)$
Layer q
Layer q-1
Layer p
Layer p-1
Correlation Matrix
$\tilde{\boldsymbol{W}}_{p\sim(q-1)} := \prod_{j=p+1}^{q-1}(\mathrm{dg}(\tilde{\boldsymbol{h}}_j)\boldsymbol{W}_j^T)\mathrm{dg}(\tilde{\boldsymbol{h}}_p)$
Forward Propagated Activation
$\tilde{\boldsymbol{x}}_{p-1}^T := \boldsymbol{x}^T\prod_{i=1}^{p-1}(\boldsymbol{W}_i\mathrm{dg}(\tilde{\boldsymbol{h}}_i))$
$\mathcal{N}_1(\alpha)$
$\mathcal{N}_2(\alpha)\setminus\mathcal{N}_1(\alpha)$
$\mathbb{I}\setminus\mathcal{N}_2(\alpha)$

(a)

(b)

Figure 3: **(a)**: Illustration of entries of the Hessian of NNs as activation paths. **(b)** The illustration of the size of the coupling set defined in asp. 3.6. The image represents the square matrix $\boldsymbol{W}$ defined at eq. (7) with its index set $\mathbb{I}$ color coded. The assumption roughly states that for the two set $\mathcal{N}_1$ (color coded by white pixels) and $\mathbb{I}\setminus\mathcal{N}_2$ (color coded by black pixels) of entries of $\boldsymbol{W}$, the correlation between $\boldsymbol{W}_{\mathcal{N}_1}$ and $\boldsymbol{W}_{\mathbb{I}\setminus\mathcal{N}_2}$ is small. Meanwhile, strong correlation is allowed in $\mathcal{N}$. Informally, each coupling set is a set of paths where patterns coded are correlated, e.g., legs, and arms of persons. Qualitatively, suppose we have a 10-layer NN with 100 neurons in each layer, then the Hessian $\boldsymbol{H}$ would be of dimension $N = 10^{10}$ (due to Kronecker product), then $N^{1/2}$ is 10^5. That is, that the coupling set could have almost 10^5 paths. **That means for each activation path, it could have almost 10^5 number of correlated activation paths among only 10^3 neurons.**

The Hessian present in eq. (5) is rather complicated. But it has a rather intuitive physical meaning. Each entry of the Hessian is a way to connect neurons in the NNs, and thus could be understood as an activation path. We explain this connection here to give an *informal* understanding of the Hessian.

As illustrated in fig. 3a, when an example $\boldsymbol{x}$ is propagated through a NN, and the gradient calculated is propagated back afterwards, for neurons in layer $p-1$, we have **activation $\tilde{\boldsymbol{x}}_{p-1}^T$**, shown as the red neuron in fig. 3a; for neurons in layer q, we have **gradient $\tilde{\boldsymbol{\alpha}}_q$**, shown as the blue neuron in fig. 3a; for neurons in layers $p, q-1$ respectively, we have their **correlation matrix $\tilde{\boldsymbol{W}}_{p\sim(q-1)}$** — are the two neurons tend to be activated simultaneously, or otherwise? — shown as the green neurons in fig. 3a. Informally, if we connect these neurons, we would obtain **an activation path** in the NN from layer $p-1$ to q, shown as the lines connecting neurons in fig. 3a. The $\otimes$ can be understood as multiplication; that is, each entry is $\tilde{\alpha}_i\tilde{w}_{jk}\tilde{x}_l$ (layer indices suppressed), and informally, the strength of the activation path. **That's to say, each entry in the Hessian $\boldsymbol{H}$ is the strength of a possible activation path that connects neurons in the NNs.**

3.4 Assumptions that characterize weak correlation in Hessian

We introduce the assumptions under a pedagogical case where the sample size m is large in eq. (1). The additional large-sample-size requirement (asp. 3.1) greatly simplifies assumptions, and enables us to describe the assumptions with more familiar concepts to the community. The theorem is proved *without* the further requirement. Asp. 3.4 3.6 in this section are simplified version of asp. A.2 A.3 under the further condition asp. 3.1. Why asp. 3.1 simplifies assumptions can be found in appendix C. The theorem is more generally held under asp. 3.2 3.3 A.2 3.5 A.3. The assumptions are from Erdos et al. [12], the paper that proposes a key technique we use. It suffices to understand for now that **when entries of the Hessian of NNs satisfy asp. 3.1 3.2 3.3 3.4 3.5 3.6, we have theorem 4.1.**

We are ready for assumptions. $\boldsymbol{H}$ is a random matrix, since it is a function of a random variable, i.e., the input sample. The assumptions characterize the statistic properties of activation paths, i.e., entries of $\boldsymbol{H}$. Let

$$\boldsymbol{A} := \mathbb{E}[\boldsymbol{H}], \frac{1}{\sqrt{N}}\boldsymbol{W} := \boldsymbol{H} - \boldsymbol{A}, \mathcal{S}[\boldsymbol{R}] := \frac{1}{N}\mathbb{E}_{\boldsymbol{W}}[\boldsymbol{W}\boldsymbol{R}\boldsymbol{W}] \tag{7}$$

where the expectation in $\mathcal{S}$ is taken w.r.t. $\boldsymbol{W}$ while keeping $\boldsymbol{R}$ fixed — it is a linear operator on the space of matrices. **Note that $\boldsymbol{W}$ is reused.** We state the assumptions as follows.

Assumption 3.1 (Large Sample Size). *$m \to \infty$, where m is the sample number in the empirical risk.*

Assumption 3.2 (Large Network Size). *$N \to \infty$. N is the dimension of the Hessian matrix $\boldsymbol{H}$.*

Asp. 3.2 is a simplification to present a non-asymptotic type result, in which a large N gives a small error bound. We briefly present an imprecise version of the bound that contains the key message here, and defer the full exposure to theorem A.1 in appendix A.2. Given the Hessian $\boldsymbol{H}$, the *resolvent* matrix $\boldsymbol{G}(z)$ (c.f. definition A.2) of $\boldsymbol{H}$ uniquely determines the probability distribution of eigenvalue of $\boldsymbol{H}$ at $\Im z$ (c.f. lemma A.1), where z is a complex number. $\boldsymbol{M}(z)$ is a matrix obtained later that characterizes the loss landscape described in section 1. The closeness of $\boldsymbol{G}$ and $\boldsymbol{M}$ is given by a high-probability bound as follows. Under asp. 3.1 3.3 3.4 3.5 3.6, for any $\epsilon > 0$, for all $D > 0$, we have

$$P(\left|\boldsymbol{x}^T(\boldsymbol{G}(z) - \boldsymbol{M}(z))\boldsymbol{y}\right| \leq ||\boldsymbol{x}||||\boldsymbol{y}||N^{\epsilon}/\sqrt{N\Im z}) \geq 1 - CN^{-D} \tag{8}$$

where $\boldsymbol{x}, \boldsymbol{y}$ are arbitrary deterministic vectors, $C > 0$ is a constant depending on D, ϵ and constants in asp. 3.4 3.6, and $\epsilon > 0$ can be chosen sufficiently small.

The bound roughly states that the probability that the eigenspectrum of the Hessian $\boldsymbol{H}$ of NNs is symmetrical grows at a rate of $1 - CN^{-D}$ w.r.t. N (note that $C > 0, D > 0$ are constants). As in generalization bounds in statistical learning theory, the exact value of the bound is not critical, while the parameters that influence the rate of the bound are more relevant. The bound implies that the benign loss landscape manifests at large N. N characterizes the size of a NN, and its value depends on both the width and depth of the NN. Compared with existing works that require astronomical [10], or infinite [23] neural layer width, the large N limits of the Hessian can be achieved by adding more layers, even though the number of neurons in each of the layers may be small compared with the overall number of neurons. As the experiments in section 1 shows, the behaviors predicted by the bound, i.e., symmetric eigenspectrum, manifest at finite N of practical NNs.

Assumption 3.3 (Zero Mean). *$\boldsymbol{A} = \boldsymbol{0}$*

Asp. 3.3 states that the mean strength of activation paths are zero (note the strength could be negative).

Asp. 3.4 and asp. 3.5 following characterize the strength of activation paths and their correlations.

Assumption 3.4 (Boundedness). *1) $\exists \mu, \sigma \in \mathbb{R}, \forall \alpha \in \mathbb{I}, \mathbb{E}[|w_\alpha|] \leq \mu, \mathbb{E}[|w_\alpha|^2] \leq \sigma$. 2) For any $\epsilon > 0, \exists C_1, C_2(\epsilon) \in \mathbb{R}$, where $C_2(\epsilon)$ depends on ϵ, $|||\kappa|||_2^{iso} \leq C_1, |||\kappa|||_2^{av} + |||\kappa|||_2^{iso} \leq C_2(\epsilon)N^{\epsilon}$*

Asp. 3.4 states that 1) the mean and variance of w_α are bounded; 2) covariance between $\boldsymbol{W}_{\mathbb{I}}$, i.e., all entries in $\boldsymbol{W}$, are bounded. The second bound requires specific norms $|||\kappa|||_2^{\text{av}}, |||\kappa|||_2^{\text{iso}}$, which we explain as follows. av stands for average, while iso stands for isotropic. $\forall \alpha, \beta \in \mathbb{I}$, denote $\alpha := (a_1, a_2), \beta := (b_1, b_2)$, and let $\boldsymbol{K}^{\text{av}}$ be the matrix, of which $\boldsymbol{K}^{\text{av}}{}_{a_1N+a_2, b_1N+b_2} := |\kappa(\alpha, \beta)|$; note that $\boldsymbol{K}^{\text{av}}$ is a $N^2 \times N^2$ matrix. $|||\kappa|||_2^{\text{av}}$ is defined as $||\boldsymbol{K}^{\text{av}}||$, the operator norm of $\boldsymbol{K}^{\text{av}}$. Note that each row of $\boldsymbol{K}^{\text{av}}$ is the covariance of w_α with all entries (including w_α itself) of $\boldsymbol{W}$. $||\boldsymbol{K}^{\text{av}}||$ characterizes a collective behavior of all the covariance. To understand the characterization, recall the definition of $||\cdot||$ on matrix as follows.

$$||\boldsymbol{K}^{\text{av}}|| = \max_{||\boldsymbol{x}||\leq 1} \boldsymbol{x}^T \boldsymbol{K}^{\text{av}} \boldsymbol{x} = \max_{||\boldsymbol{x}||\leq 1} \sum_{a_1=1,b_1=1,a_2=1,b_2=1}^{a_1=N,b_1=N,a_2=N,b_2=N} x_{a_1 N+b1} \left|\kappa(a_1 b_1, a_2 b_2)\right| x_{a_2 N+b2}.$$

Thus, it is the largest possible weighted average of absolute covariance. Asp. 3.4 puts a cap on the largest weighted average. $|||\kappa|||_2^{\text{iso}}$ defines a similar quantity with $|||\kappa|||_2^{\text{av}}$. We defer its definition to definition A.7, so not to clutter the message with technical definitions too much.

Assumption 3.5 (Comparibility)**.** $\exists 0 < c < C, \forall \boldsymbol{T} \succ \boldsymbol{0}, c\, N^{-1} tr\, \boldsymbol{T} \preceq \mathcal{S}[\boldsymbol{T}] \preceq C N^{-1} tr\, \boldsymbol{T}$.

Asp. 3.5 is a certain mean field condition that makes the covariance that does exist between entries in $\boldsymbol{W}$ comparable. To see how, we write $\mathcal{S}[\boldsymbol{T}]_{pq} = 1/N \sum_{i,j=1}^{N} t_{ij} \mathbb{E}[w_{pi} w_{jq}]$. Let $s_{pq} := \mathcal{S}[\boldsymbol{T}]_{pq}$. Thus, for a given $\boldsymbol{T}$, s_{pq} is a weighted summation of a set of covariance between $\boldsymbol{W}_{:p}$ and $\boldsymbol{W}_{:q}$. Note that the weights t_{ij} are the same for all p, q. Similarly with the explanation of asp. 3.4, $c\, N^{-1}\text{tr}\, \boldsymbol{T} \preceq \mathcal{S}[\boldsymbol{T}] \preceq C N^{-1}\text{tr}\, \boldsymbol{T}$ asks the smallest and the largest weighted average of entries of $\mathcal{S}[\boldsymbol{T}]$ to be lower and upper bounded by $c\, N^{-1}\text{tr}\, \boldsymbol{T}, C N^{-1}\text{tr}\, \boldsymbol{T}$ respectively. If the largest weighted average is up to C/c times larger than the smallest, it implies s_{pq} are comparable; that is, different sets of covariance involved in s_{pq} are comparable in term of the summation arbitrated by any $\boldsymbol{T} \succeq \boldsymbol{0}$. To see an example, let T to be a diagonal matrix with a single non-zero entry at (i,i), we have $\mathcal{S}[\boldsymbol{T}]_{pq} = (1/N) t_{ii} \mathbb{E}[w_{pi} w_{iq}]$. Further letting $\mathbb{E}[w_{pi} w_{iq}] = 0, p \neq q$, $\mathcal{S}[\boldsymbol{T}]$ is thus a diagonal matrix with diagonal elements $s_{pp} = (t_{ii}/N) \mathbb{E}[w_{pi}^2]$ (recall $\boldsymbol{W}$ is symmetric). The condition asks the variance $\mathbb{E}[w_{pi}^2], p = 1 \ldots N$ to be comparable (in the the band $[c, C]$).

Assumption 3.6 (Diversity)**.** *There exists $\mu > 0$ such that the following holds: for every $\alpha \in \mathbb{I}$, there exists two sets $\mathcal{N}_k = \mathcal{N}_k(\alpha), k = 1, 2$ such that $\alpha \in \mathcal{N}_1 \subset \mathcal{N}_2 = \mathcal{N} \subset \mathbb{I}$, $|\mathcal{N}| \leq N^{1/2-\mu}$ and*

$$\kappa(f(\boldsymbol{W}_{\mathbb{I}\setminus\mathcal{N}}), g(\boldsymbol{W}_{\mathcal{N}_1})) \leq N^{-3} ||f||_3 ||g||_3,$$

for any real analytic functions f, g, where $||||_3$ is the L^3 norm on function space. We call the set $\mathcal{N}$ of α the **coupling set** *of α.*

Asp. 3.6 divides entries of $\boldsymbol{W}$ into two sets, and states that the covariance between these two sets should be weak. The weakness can be seen by comparing it with the boundedness asp. 3.4, where $|||\kappa|||^{\text{av}}$ asks the averaged covariance between w_α and $\boldsymbol{W}_{\mathbb{I}}$ not to exceed a upper bound, i.e., C_1, while the diversity assumption asks their covariance should decay at cubic rate w.r.t. N. Note that the boundedness assumption actually characterizes how strong the correlation is if there is correlation. Given a w_α, the diversity assumption actually restricts the correlation to a relatively small set, i.e. the coupling set $\mathcal{N}$. To give an intuitive feeling about the size of $\mathcal{N}$, we illustrate it with fig. 3b.

3.5 Assumptions and initialization/normalization techniques in practice

As shown in section 1, the landscape prescribed by theorem 4.1 is observed in practical NNs. We conjecture that the set of assumptions are also satisfied by the practical NNs. We offer a sketch of intuition why these assumptions might be satisfied. The sketch serves two goals: 1) an more detailed *informal* explanation on the intuition of the assumptions than the one in the beginning of section 3 to help understanding; 2) a sketch that outlines the logic of future experiments that might verify that the assumptions are actually satisfied in practical NNs.

We remark on assumptions that does not relate with normalization briefly first. As discussed, asp. 3.1 is intended to simplify the assumptions to convey the key ideas, and is not necessary. Asp. 3.2 is an non-asymptotic description that requires NNs to be large. The larger the network, the higher the probability that the eigenspectrum of Hessian is symmetric, and thus contains negative eigenvalues for further optimization. As shown in fig. 1b, even the smallest practical NNs, i.e., LeNet, is large enough to let the symmetry emerge, since normally the dimension N of Hessian is very large even for small networks, c.f. N in the caption of fig. 3b.

Thus, the key assumptions to be satisfied are asp. 3.3 3.4 3.5 3.6. The assumptions are intimately with *normalization/initialization* techniques that normalize data, activation, weights to be zero-mean and almost unit variance. We discuss them one by one.

- **Asp. 3.3.** As explain in section 3.3, the Hessian consists of block matrices $\boldsymbol{H}_{pq} = \tilde{\boldsymbol{\alpha}}_q \otimes \tilde{\boldsymbol{W}}_{p\sim(q-1)} \otimes \tilde{\boldsymbol{x}}_{p-1}^T$. Batch normalization normalizes the mean of $\tilde{\boldsymbol{x}}_{p-1}^T$ to be zero throughout training. Initialization schemes [15] initializes each entry in weight matrices in $\tilde{\boldsymbol{W}}_{p\sim(q-1)}$ to be zero-mean. Normally mean is subtracted from data before feeding in the network for training. These techniques likely makes asp. 3.3 approximately satisfied. It is interesting to explore their connection formally, or even formulate new techniques to enforce asp. 3.3 precisely.

- **Asp. 3.4.** Note assumption 3.4 that asks the correlation between entries of Hessian is bounded is supposed to be held for any N uniformly. Here, correlation includes the correlation with oneself, e.g., variance. By normalizing the activation to be of unit variance, the correlations, e.g., the variance $\mathbb{E}[|w_\alpha|^2]$, and covariance norms $|||\kappa|||_2^{\text{iso}}, |||\kappa|||_2^{\text{av}}$, are made possible to be bounded as the NN becomes larger/deeper, as explained in the following. Suppose that no normalization is enforced, and initialization is not properly designed. The exponential explosion of activation/gradient would indefinitely increase the absolute value of the correlation, e.g., $\mathbb{E}[|w_\alpha w_\beta|]$, between entries that have larger variance (as a result of exponential amplification), and the rest of the layers. Consequently, the correlation would grow with N, and thus violates asp. 3.4 — e.g., the variance $\mathbb{E}[|w_\alpha|^2]$ is not bounded anymore. This also fits well with the phenomenon that deeper NNs without normalization are harder to optimize. Through normalization, the absolute value of correlation between Hessian entries becomes meaningful/comparable, and the strong correlation between entries can only emerge from a *randomly initialized NN* through training that probably captures the composition relationship (c.f. discussion in the beginning of section 3). Normalization makes the precondition possible that absolute value of correlation within entries does not grow with N, and the moments and the norm that characterizes the "average" strength of correlation in asp. 3.4 could be bounded for any N.
- **Asp. 3.5**. The upper bound in asp. 3.5 is similarly made possible with that in asp. 3.4. We explain how the lower bound is made possible as well. The rationale is similar, except that in this case, it is the exponentially varnishing that matters. As an extreme example, all the assumptions but the lower bound in asp. 3.5 can be satisfied by a zero matrix. Thus, some requirement on the smallness of the correlation should also exist to ensure that meaningful information exists. Again, suppose that no normalization, or the network is not properly initialized. The exponential varnishing of activation/gradient would indefinitely descrease the absolute value of the correlation, e.g., $\mathbb{E}[|w_\alpha w_\beta|]$, between entries that have smaller variance (as a result of such exponential shrinking), and the rest of the layers. Consequently, the correlation would decrease with N, and thus the majority of $\mathbb{E}[|w_\alpha w_\beta|], \alpha, \beta \in \mathbb{I}$ (for a sufficiently large N) would be close to zero. It would make covariance not comparable anymore, and violate the lower bound in asp. 3.5. Through normalization, the absolute value of correlation between Hessian entries becomes meaningful/comparable, and the small correlation between entries could possibly meaningfully represent uncorrelatedness. Thus, it prevents NNs from degenerating into attractors in the hypothesis space that does not contain meaningful hypotheses. Normalization makes the precondition possible that absolute value of correlation within entries does not shrink with N, and the lower bound in asp. 3.5 could be satisfied.
- **Asp. 3.6**. Similarly with the connection between normalization and asp. 3.4, the exponentially amplified entries of Hessian would have a high correlation with the rest of all entries, and thus violate asp. 3.6, which asks each entry to only correlate with a small subset of entries. Through normalization, at random initialization, the *randomness* preconditions the network to have almost no correlations between entries of Hessian. As long as the correlation created during training does not violate asp. 3.6, theorem 4.1 would hold, and the risk can be further minimized. We remark at the caption of fig. 3b that the correlation that could exist is non-trivial.

To sum up, at initialization, normalization of mean and variance avoids exponential explosion or varnishing of w_α, and preconditions NNs to have random w_αs in $\boldsymbol{W}_{\mathbb{I}}$ that almost have no correlation with each other — no correlation does not equate with the fact that NNs have no discriminative power, since feeding data into random NN is to create random projection of data, which is known to preserve distance [3]. We conjecture normalization and random initialization techniques widely used make asp. 3.3 3.4 3.5 3.6 satisfied. As long as the correlation emerged from training does not exceed a threshold (which is non-trivial as explained in the caption of fig. 3b), theorem 4.1 holds, and zero risk can be reached. The verification of the phenomenon described in the above bullet pointed list will be future works.

4 Main results and the sketch of its proof

We present our main result in this section. We first present the main result, theorem 4.1, in section 4.1, then the remaining of the key ideas in the proof of theorem 4.1 in section 4.2, and lastly the proof sketch in section 4.3 that draws the connection between the ideas introduced in section 2 and 3 and the ideas introduced here.

4.1 Main results

Theorem 4.1. *Let $R_m(T)$ be the risk function (defined at eq. (1)) of a NN T having only one output neuron (defined at eq. (3)) with a loss function l of class $\mathcal{L}_0$ (defined in section 3.1). If the Hessian $\boldsymbol{H} \in \mathbb{R}^{N\times N}$ (c.f. eq. (6)) of $R_m(T)$ satisfies assumptions 3.2 3.3 A.2 3.5 A.3, then for $R_m(T)$,*

a) all local minima are global minima with zero risk.

b) all non-zero-risk stationary points are saddle points with half of the non-zero eigenvalues negative.

c) A constant $\lambda_0 \in \mathbb{R}$ exists, such that $||\boldsymbol{H}||$ is upper bounded by $\mathbb{E}_m[l'(T(X), Y)]\lambda_0$, where $\mathbb{E}_m[l'(T(X), Y)]$ is the empirical expectation of derivative l' of l. It implies the eigen-spectrum of $\boldsymbol{H}$ is increasingly concentrated around zero as more examples have the zero loss (classified correctly in term of surrogate loss) — since when the loss $l(\boldsymbol{x}, y)$ of an example is zero, its derivative at $\boldsymbol{x}$ is zero, i.e., $l'(\boldsymbol{x}, y) = 0$ (due to the properties of $\mathcal{L}_0$). Thus, the regions around the minima are flat basins.

The theorem is explained in section 1.

4.2 Random matrices with correlations

Section 2 and 3 show that NNs are inherently stochastic, thus the Hessian of the loss function is a real symmetric random matrix. The assumptions present in section 3.4 characterizes a type of weak correlation in the random matrix, which if satisfied, would induce the *concentration of measure* phenomenon that makes the eigen-spectrum of the random matrix symmetrically distributed around zero. This is the last key idea that leads to theorem 4.1.

The core idea of concentration of measure has all been contained in the classic normal distribution. Given an ensemble of i.i.d. r.v.s centering at 0, since the r.v.s are independent with each other, and have an equal probability being positive, or negative, they tend to cancel each other out when averaging their values, symmetrically spread around 0, and approach a normal distribution. Thus, sampling many samples from the average of the ensemble, 95% of them would fall in the range between $[+2\sigma, 2\sigma]$, where σ is the standard deviation of the ensemble.

What we need is the concentration of measure in the space of matrices. For $1 \leq i < j < \infty$, let a_{ij} be i.i.d. r.v.s. with mean 0 and variance 1, and $a_{ij} = a_{ji}$. Then $\boldsymbol{A}_N = [a_{ij}]_{i,j=1}^N$ is a matrix with random entries. It is the classic random matrix named *Wigner matrix*. Given an eigenvector $\boldsymbol{t}$ of $\boldsymbol{A}_N$, we know that when we multiply it with $\boldsymbol{A}$, we have $(\boldsymbol{At})_i = \sum_k a_{ik}t_k$. Note that when $\{a_{ik}t_k\}_{k=1,\ldots,N}$ are i.i.d. r.v.s, $a_{ik}t_k$ also tend to cancel each other out and spread symmetrically around 0, resulting in a close-to-zero eigenvalue. Thus, when the dimension of the matrix N is large enough, we would have the eigen-spectrum of the distribution shown in fig. 2b. It is termed *semi-circle law*. The eigen-spectrum is obtained by solving a *self-consistent* equation as following

$$1 + (z + m)m = 0, \Im m > 0, \Im z > 0$$

where z is a constant complex number, m is the unknown complex number to solve. Then, we apply *inverse stieltjes transform* (c.f. lemma A.1) to get the spectrum.

Note that Wigner matrix has very strong requirements on its entries: independent identically distributed. Previous works of Choromanska et al. [6] and Pennington and Bahri [40] are partly based on Wigner matrix, thus fall short on realisticity. However, notice that the key phenomenon we need is just the concentration of measure, a.k.a., $\{a_{ik}t_k\}_{k=1,\ldots,N}$ tend to cancel out and symmetrically spread around 0. Thus, the conditions we need are just they are of weak correlation with each other to obtain a semi-circle-law-like eigen-spectrum. The assumptions introduced in section 3.4 aim to characterize such a condition. **Under asp. A.1 A.2 3.5 A.3, Erdos et al. [12] obtain the eigen-spectrum of a symmetric random matrix by solving a matrix version of the above scalar equation**, termed *Matrix Dyson Equation (MDE)*

$$\boldsymbol{I} + (z - \boldsymbol{A} + \mathcal{S}[\boldsymbol{G}])\boldsymbol{G} = \boldsymbol{0}, \Im \boldsymbol{G} \succ \boldsymbol{0}, \Im z > 0, \tag{9}$$

where $\Im \boldsymbol{G} \succ 0$ means $\Im \boldsymbol{G}$ is positive definite, and the rest of the symbols are defined at eq. (7). The details on how the equation is derived, and how it is related to eigen-spectrum are explained in appendix A.2. We only aim to brief its intuition in this section.

4.3 Proof sketch of theorem 4.1: landscape analysis of NNs

The main content of the paper is actually the proof sketch. With all the key ideas introduced, we connect these ideas to present the proof sketch of theorem 4.1.

1. First, we show that NNs are hierarchical measure transports in section 2, so the Hessian $\boldsymbol{H}$ of a NN is a random matrix.
2. Second, in section 3.2, for a restricted class $\mathcal{L}_0$ of loss functions, by calculating the Hessian, we show that it is a real symmetric matrix. The actual calculation of the Hessian is carried out in appendix A.1.
3. Third, if the Hessian satisfies the weak correlation characterized in section 3.4, as a result of the concentration of measure phenomenon introduced in section 4.2, the eigenspectrum of $\boldsymbol{H}$ can be obtained by solving the Matrix Dyson Equation. MDE is formally introduced in appendix A.2.
4. Lastly, we can proceed to analyze the eigen-spectrum of Hessian of NNs, which is carried out in appendix A.3. Though the eigen-spectrum cannot be obtained in closed-form as semi-circle law in the case of Wigner matrix, we still can analyze it behaviors through MDE. In **theorem A.2**, we prove the solutions to MDE is symmetry w.r.t. y-axis under the assumptions of theorem 4.1: that is, given any positive eigenvalue λ, $-\lambda$ is also an eigenvalue. Since there are always non-zero eigenvalues by theorem A.1, we always have negative eigenvalues. We break down the analysis of Hessian $\boldsymbol{H}$ into two cases: 1) for all training samples, at least one sample (x, y) has non-zero loss value; 2) and all training samples are classified properly with zero loss values. In case 1), by theorem A.2, $\boldsymbol{H}$ is always indefinite, thus corresponds to a saddle point. In case 2), we are at global optima, where all losses are zero. Lastly, theorem 4.1.c is proved, which is straightforward.

References

[1] Johannes Alt, Laszlo Erdos, and Torben Krüger. The Dyson equation with linear self-energy: spectral bands, edges and cusps. Technical report, 2018. URL http://arxiv.org/abs/1804.07752.

[2] Alexandr Andoni, Rina Panigrahy, Gregory Valiant, and Li Zhang. Learning polynomials with neural networks. In *ICML*, 2014.

[3] Fabio Anselmi, Lorenzo Rosasco, and Tomaso Poggio. On Invariance and Selectivity in Representation Learning. *Information and Inference, Special Issue: Deep Learning*, may 2016.

[4] Alon Brutzkus and Amir Globerson. Globally Optimal Gradient Descent for a ConvNet with Gaussian Inputs. In *ICML*, 2017.

[5] Lenaic Chizat and Francis Bach. On the Global Convergence of Gradient Descent for Over-parameterized Models using Optimal Transport. In *NIPS*, 2018.

[6] Anna Choromanska, Mikael Henaff, and Michael Mathieu. The Loss Surfaces of Multilayer Networks. In *AISTATS*, 2015.

[7] Anna Choromanska, Yann LeCun, and Gérard Ben Arous. Open Problem: The landscape of the loss surfaces of multilayer networks. In *COLT*, 2015.

[8] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). In *ICLR*, 2015.

[9] Simon S. Du and Jason D. Lee. On the Power of Over-parametrization in Neural Networks with Quadratic Activation. In *ICML*, 2018.

[10] Simon S. Du, Jason D. Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient Descent Finds Global Minima of Deep Neural Networks. In *ICML*, 2019.

[11] N. Dunford and J.T. Schwartz. *Linear operators. Part 1: General theory*. Pure and Applied Mathematics. Interscience Publishers, 1957.

[12] Laszlo Erdos, Torben Kruger, and Dominik Schroder. Random Matrices with Slow Correlation Decay. *Forum of Mathematics, Sigma*, 7(8), 2019. doi: doi:10.1017/fms.2019.2.

[13] Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An Investigation into Neural Net Optimization via Hessian Eigenvalue Density. In *ICML*, 2019.

[14] Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Limitations of Lazy Training of Two-layers Neural Networks. Technical report, 2019. URL `http://arxiv.org/abs/1906.08899`.

[15] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.

[16] Uffe Haagerup and Steen Thorbjørnsen. A new application of random matrices:. *Annals of Mathematics*, 162(2):711–775, 2005. ISSN 0003-486X. doi: 10.4007/annals.2005.162.711.

[17] Benjamin D. Haeffele and René Vidal. Global optimality in neural network training. In *CVPR*, 2017.

[18] Moritz Hardt and Tengyu Ma. Identity Matters in Deep LearningIdentity Matters in Deep Learning. In *ICLR*, 2017.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *ICCV*, 2015.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016.

[21] J. William Helton, Reza Rashidi Far, and Roland Speicher. Operator-valued semicircular elements: Solving a quadratic matrix equation with positivity constraints. *International Mathematics Research Notices*, 2007:1–14, 2007. ISSN 10737928. doi: 10.1093/imrn/rnm086.

[22] Christian Ioffe, Sergey and Szegedy. Batch Normalization : Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *ICML*, 2015.

[23] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural Tangent Kernel: Convergence and Generalization in Neural Networks. In *NIPS*, 2018.

[24] L.P. Kadanoff. *Statistical Physics: Statics, Dynamics and Renormalization*. Statistical Physics: Statics, Dynamics and Renormalization. World Scientific, 2000. ISBN 9789810237646.

[25] Kenji Kawaguchi. Deep Learning without Poor Local Minima. In *NIPS*, 2016.

[26] John E Kelly. Computing, cognition and the future of knowing. Technical report, 2015.

[27] A. Klenke. *Probability Theory: A Comprehensive Course*. World Publishing Corporation, 2012. ISBN 9787510044113.

[28] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS*, 2012.

[29] Thomas Laurent and James von Brecht. Deep linear neural networks with arbitrary loss: All local minima are global. In *ICML*, 2018.

[30] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. ISSN 00189219. doi: 10.1109/5.726791.

[31] Yuanzhi Li and Yang Yuan. Convergence Analysis of Two-layer Neural Networks with ReLU Activation. In *NIPS*, 2017.

[32] Shiyu Liang, Ruoyu Sun, Yixuan Li, and R Srikant. Understanding the Loss Surface of Neural Networks for Binary Classification. In *ICML*, 2018.

[33] Lin Lin, Yousef Saad, and Chao Yang. Approximating spectral densities of large matrices. *SIAM review*, 58(1):34–65, 2016.

[34] J.R. Magnus and H. Neudecker. *Matrix differential calculus with applications in statistics and econometrics: 3rd. ed.* John Wiley & Sons, Limited, 2007.

[35] P. McCullagh. *Tensor methods in statistics*. Monographs on statistics and applied probability. Chapman and Hall, 1987. ISBN 9780412274800.

[36] Quynh Nguyen and Matthias Hein. The loss surface of deep and wide neural networks. In *ICML*, 2017.

[37] Quynh Nguyen and Matthias Hein. Optimization Landscape and Expressivity of Deep CNNs. In *ICML*, 2018.

[38] Quynh Nguyen, Mahesh Chandra Mukkamala, and Matthias Hein. On the loss landscape of a class of deep neural networks with no bad local valleys. In *ICLR*, 2019.

[39] Vardan Papyan. The Full Spectrum of Deep Net Hessians At Scale: Dynamics with Sample Size. Technical report, 2018. URL http://arxiv.org/abs/1811.07062.

[40] Jeffrey Pennington and Yasaman Bahri. Geometry of Neural Network Loss Surfaces via Random Matrix Theory. In *ICML*, 2017.

[41] F W Pfeiffer. Automatic differentiation in prose. In *ICLR Workshop*, 2017.

[42] Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for Activation Functions. Technical report, oct 2017. URL http://arxiv.org/abs/1710.05941.

[43] Itay Safran and Ohad Shamir. On the Quality of the Initial Basin in Overspecified Neural Networks. In *ICML*, 2016.

[44] Hanie Sedghi and Anima Anandkumar. Provable Methods for Training Neural Networks with Sparse Connectivity. In *ICLR Workshop*, 2014.

[45] N Siddharth, Brooks Paige, Alban Desmaison, Frank Wood, Philip Torr, and Noah D Goodman. Learning deep models: Critical points and local openness. In *ICLR Workshop*, 2018.

[46] Mahdi Soltanolkotabi. Learning ReLUs via Gradient Descent. In *NIPS*, 2017.

[47] Mahdi Soltanolkotabi, Adel Javanmard, and Jason D Lee. Theoretical insights into the optimization landscape of over-parameterized shallow neural networks. *IEEE Transactions on Information Theory*, 4(8), 2018. doi: 10.1109/TIT.2018.2854560.

[48] Daniel Soudry and Elad Hoffer. Exponentially vanishing sub - optimal local minima in multilayer neural networks. In *ICLR Worksop*, 2018.

[49] T. Tao. *Topics in Random Matrix Theory*. Graduate studies in mathematics. American Mathematical Soc., 2012. ISBN 9780821885079.

[50] Luca Venturi, Afonso S. Bandeira, and Joan Bruna. Spurious Valleys in Two-layer Neural Network Optimization Landscapes. Technical report, 2018. URL http://arxiv.org/abs/1802.06384.

[51] E. P. Wigner. Statistical properties of real symmetric matrices with many dimensions.pdf. In *Canadian Mathematical Congress Proceedings*, 1957.

[52] Chulhee Yun, Suvrit Sra, and Ali Jadbabaie. Global optimality conditions for deep neural networks. In *ICLR*, 2018.

[53] Kai Zhong, Zhao Song, Prateek Jain, Peter L. Bartlett, and Inderjit S. Dhillon. Recovery Guarantees for One-hidden-layer Neural Networks. In *ICML*, 2017.

Appendices

A Full proof of theorem 4.1

We present the full proof of theorem 4.1 in this section, the sketch of which has been outline in section 4.3. In appendix A.1, we derive the Hessian of NNs, of which the intuition is explained in section 3.1-3.3. In appendix A.2, we introduce the random matrix tool we used, i.e., Matrix Dyson Equation (MDE), to study the Hessian, of which the intuition is explained in section 4.2. In appendix A.3, we use MDE to study the loss landscape of NNs, and prove theorem 4.1 and results that lead to it. The sketch of appendix A.3 is present in section 4.3 as well.

A.1 Hessian of NN is inherently a huge random matrix

As explained, to study the landscape of the loss function, we study the eigenvalue distribution of its Hessian $\boldsymbol{H}$ at the critical points. First, we derive the Hessian $\boldsymbol{H}$ of loss function of class $\mathcal{L}_0$ composed upon NNs with a single output. For a review of matrix calculus, the reader may refer to Magnus and Neudecker [34].

The first partial differential of l defined at eq. (4) w.r.t. $\boldsymbol{W}_p$ is

$$\begin{aligned}\partial l(T\boldsymbol{x}, y) = l'(T\boldsymbol{x}, y)\boldsymbol{\alpha}^T \prod_{j=p+1}^{L-1} (\text{dg}(\tilde{\boldsymbol{h}}_j)\boldsymbol{W}_j^T)\text{dg}(\tilde{\boldsymbol{h}}_p) \\ \otimes \boldsymbol{x}^T \prod_{i=1}^{p-1}(\boldsymbol{W}_i\text{dg}(\tilde{\boldsymbol{h}}_i))\partial\text{vec}\boldsymbol{W}_p\end{aligned}$$

where $\otimes$ denotes Kronecker product, and $\{\tilde{\boldsymbol{h}}_i\}_{i=1,\ldots,L-1}$ are realization of r.v.s H defined at eq. (2). Note that for clarity of presentation, we use the partial differential the same way as differential is defined and used in Magnus and Neudecker [34], i.e., ∂l is a number instead of an infinitely small quantities, though in the book, partial differential is not defined explicitly.

Since $\boldsymbol{H}$ is symmetric, we only need to compute the block matrices by taking partial differential w.r.t. $\boldsymbol{W}_q$, where $q > p$ — taking partial differential w.r.t. $\boldsymbol{W}_p$ again gives zero matrix.

$$\begin{aligned}&\partial^2 l(T\boldsymbol{x}, y) \\ =&l'(T\boldsymbol{x}, y)[(\boldsymbol{\alpha}^T \prod_{k=q+1}^{L-1} (\text{dg}(\tilde{\boldsymbol{h}}_k)\boldsymbol{W}_k^T)\text{dg}(\tilde{\boldsymbol{h}}_q) \\ \otimes&\text{dg}(\tilde{\boldsymbol{h}}_p) \prod_{j=p+1}^{q-1} (\boldsymbol{W}_j\text{dg}(\tilde{\boldsymbol{h}}_j))\partial\text{vec}\boldsymbol{W}_q)^T \\ \otimes&\boldsymbol{x}^T \prod_{i=1}^{p-1}(\boldsymbol{W}_i\text{dg}(\tilde{\boldsymbol{h}}_i))]\partial\text{vec}\boldsymbol{W}_p\end{aligned}$$

$$\begin{aligned}=&l'(T\boldsymbol{x}, y) \\ &(\partial\text{vec}\boldsymbol{W}_q)^T[\text{dg}(\tilde{\boldsymbol{h}}_q) \prod_{k=q+1}^{L-1} (\boldsymbol{W}_k\text{dg}(\tilde{\boldsymbol{h}}_k))\boldsymbol{\alpha} \\ \otimes& \prod_{j=p+1}^{q-1} (\text{dg}(\tilde{\boldsymbol{h}}_j)\boldsymbol{W}_j^T)\text{dg}(\tilde{\boldsymbol{h}}_p) \\ \otimes&\boldsymbol{x}^T \prod_{i=1}^{p-1}(\boldsymbol{W}_i\text{dg}(\tilde{\boldsymbol{h}}_i))]\partial\text{vec}\boldsymbol{W}_p\end{aligned}$$

Thus, $\boldsymbol{H}$ is an ensemble of real symmetric random matrix with correlated entries. Denote

$$\begin{aligned}\boldsymbol{H}_{pq} =& l'(T\boldsymbol{x}, y)\mathrm{dg}(\tilde{\boldsymbol{h}}_q) \prod_{k=q+1}^{L-1} (\boldsymbol{W}_k \mathrm{dg}(\tilde{\boldsymbol{h}}_k))\boldsymbol{\alpha} \\ & \otimes \prod_{j=p+1}^{q-1} (\mathrm{dg}(\tilde{\boldsymbol{h}}_j)\boldsymbol{W}_j^T)\mathrm{dg}(\tilde{\boldsymbol{h}}_p) \\ & \otimes \boldsymbol{x}^T \prod_{i=1}^{p-1} (\boldsymbol{W}_i \mathrm{dg}(\tilde{\boldsymbol{h}}_i)) \end{aligned} \tag{10}$$

We have the Hessian of l as

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{H}_{12}^T & \dots & \boldsymbol{H}_{1L}^T \\ \boldsymbol{H}_{12} & \boldsymbol{0} & \dots & \boldsymbol{H}_{2L}^T \\ \vdots & \ddots & \ddots & \vdots \\ \boldsymbol{H}_{1L} & \boldsymbol{H}_{2L} & \dots & \boldsymbol{0} \end{bmatrix} \tag{11}$$

Equation (10) could be reformulated, so it may become easier to comprehend, as we explain in section 3.3. We rewrite eq. (10) in the following form on the right, where on the left are shorthanded symbols defined for the reformulation (the equation should be read vertically)

$$\begin{aligned} \tilde{\boldsymbol{W}}_{q\sim(L-1)} &:= \mathrm{dg}(\tilde{\boldsymbol{h}}_q) \prod_{k=q+1}^{L-2} (\boldsymbol{W}_k \mathrm{dg}(\tilde{\boldsymbol{h}}_k))\boldsymbol{W}_{L-1} \\ \tilde{\boldsymbol{\alpha}}_q &:= \tilde{\boldsymbol{W}}_{q\sim(L-1)}\mathrm{dg}(\tilde{\boldsymbol{h}}_{L-1})\boldsymbol{\alpha} l'(T\boldsymbol{x}, y) \\ \tilde{\boldsymbol{W}}_{p\sim(q-1)} &:= \prod_{j=p+1}^{q-1} (\mathrm{dg}(\tilde{\boldsymbol{h}}_j)\boldsymbol{W}_j^T)\mathrm{dg}(\tilde{\boldsymbol{h}}_p) \\ \tilde{\boldsymbol{W}}_{1\sim(p-1)} &:= \prod_{i=1}^{p-1} (\boldsymbol{W}_i \mathrm{dg}(\tilde{\boldsymbol{h}}_i)) \\ \tilde{\boldsymbol{x}}_{p-1}^T &:= \boldsymbol{x}^T \tilde{\boldsymbol{W}}_{1\sim(p-1)} \end{aligned}$$

$$\boldsymbol{H}_{pq} = l'(T\boldsymbol{x}, y)\tilde{\boldsymbol{W}}_{q\sim(L-1)}\mathrm{dg}(\tilde{\boldsymbol{h}}_{L-1})\boldsymbol{\alpha} \quad = \tilde{\boldsymbol{\alpha}}_q \tag{12a}$$

$$\otimes \tilde{\boldsymbol{W}}_{p\sim(q-1)} \qquad \otimes \tilde{\boldsymbol{W}}_{p\sim(q-1)} \tag{12b}$$

$$\otimes \boldsymbol{x}^T \tilde{\boldsymbol{W}}_{1\sim(p-1)} \qquad \otimes \tilde{\boldsymbol{x}}_{p-1}^T \tag{12c}$$

Through such rewriting, entries of Hessian can correspond to components in activation paths, as discussed in section 3. The connection is illustrated in fig. 3a. Drawn in blue, eq. (12a) is a vector $\tilde{\boldsymbol{\alpha}}_q$ created by back propagating the vector $\mathrm{dg}(\tilde{\boldsymbol{h}}_{L-1})\boldsymbol{\alpha} l'(T\boldsymbol{x}, y)$ to layer q by left multiplying $\tilde{\boldsymbol{W}}_{q\sim(L-1)}$— note that it is the *back propagated gradient*. Drawn in green, eq. (12b) is the *covariance matrix without removing the mean* between neurons at layer p and layer $q-1$, when taking expectation w.r.t. samples, i.e., $\mathbb{E}_{z\sim\mu^{\mathcal{Z}}}[\tilde{\boldsymbol{W}}_{p\sim(q-1)}]$ (it could be understood as the strength of correlation between the neurons at layer p and $q-1$). Drawn in red, eq. (12c) is the *forward propagated activation* at layer $p-1$.

Recall that the objective function is the empirical risk function $R_m(T)$ at eq. (1). With the Hessian of loss function at an example $\boldsymbol{x}$ is calculated. Given a set of i.i.d. training samples $(X_i, Y_i)_{i=1,\dots,m}$, $R_m(T)$ is a summation of the i.i.d. random matrices. Formally, reusing the notation to denote $\boldsymbol{H}$ the Hessian of $R_m(T)$ and $\boldsymbol{H}_i$ the Hessian of $l(T(X_i;\boldsymbol{\theta}), Y_i)$, we have

$$\boldsymbol{H} = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{H}_i \tag{13}$$

A.2 Eigenvalue distribution of symmetry random matrix with slow correlation decay

In this section, we show how to obtain the eigen-spectrum of $\boldsymbol{H}$ through random matrix theory (RMT) [49]. RMT has been born out of the study on the nuclei of heavy atoms, where the spacings between lines in the spectrum of a heavy atom nucleus is postulated to be the same with spacings between eigenvalues of a random matrix [51]. In a certain way, it is one of the backbone math of complex systems, where the collective behaviors of sophisticated subunits can be analyzed stochastically when deterministic or analytic analysis is intractable.

The following definitions can be found in Tao [49] unless otherwise noted.

The eigen-spectrum is studied as empirical spectral distribution (ESD) in RMT, define as

Definition A.1 (Empirical Spectral Distribution)**.** *Given a $N \times N$ random matrix $\boldsymbol{H}$, its* **empirical spectral distribution** *$\mu_{\boldsymbol{H}}$ is*

$$\mu_{\boldsymbol{H}} = \frac{1}{N}\sum_{i=1}^{N} \delta_{\lambda_i}$$

where $\{\lambda_i\}_{i=1,\ldots,N}$ are all eigenvalues of $\boldsymbol{H}$ and δ is the delta function.

Given a hermitian matrix $\boldsymbol{H}$, its ESD $\mu_{\boldsymbol{H}}(\lambda)$ can be studied via its resolvent $\boldsymbol{G}$.

Definition A.2 (Resolvent)**.** *Let $\boldsymbol{H}$ be a normal matrix, and $z \in \mathbb{H}$ a spectral parameter. The* **resolvent** *$\boldsymbol{G}$ of $\boldsymbol{H}$ at z is defined as*

$$\boldsymbol{G} = \boldsymbol{G}(z) = \frac{1}{\boldsymbol{H} - z}$$

where

$$\mathbb{H} := \{z \in \mathbb{C} : \Im z > 0\}$$

$\mathbb{C}$ denotes the complex field, and $\Im$ is the function gets imaginary part of a complex number z.

$\boldsymbol{G}$ compactly summarizes the spectral information of $\boldsymbol{H}$ around z, which is normally analyzed by *functional calculus* on operators, and defined through Cauchy integral formula on operators as follows.

Definition A.3 (Functions of Operators)**.**

$$f(T) = \frac{1}{2\pi i}\int_C \frac{f(\lambda)}{\lambda - T} d\lambda \tag{14}$$

where f is an analytic scalar function and C is an appropriately chosen contour in $\mathbb{C}$.

The formula can be defined on a range of linear operators [11] (Recall that a linear operator is a linear mapping whose domain and codomain are defined on the same field). Since the most complex case involved here will be a normal matrix, we stop at stating that the formula holds true when T is a normal matrix.

Resolvent $\boldsymbol{G}$ is related to eigen-spectrum of $\boldsymbol{H}$ through stieltjes transform of $\mu_{\boldsymbol{H}}(\lambda)$.

Definition A.4 (Stieltjes Transform)**.** *Let μ be a Borel probability measure on $\mathbb{R}$. Its* **Stieltjes transform** *at a spectral parameter $z \in \mathbb{H}$ is defined as*

$$m_\mu(z) = \int_{\mathbb{R}} \frac{d\mu(x)}{x - z}$$

With some efforts, it can be seen that the normalized trace of $\boldsymbol{G}$ is stieltjes transform of eigen-spectrum of $\boldsymbol{H}$

$$m_{\mu_{\boldsymbol{H}}}(z) = \frac{1}{N}\mathrm{tr}\, G$$

For a proof, the reader may refer to proposition 2.1 in Alt et al. [1]. $\mu_{\boldsymbol{H}}$ can be recovered from $m_{\mu_{\boldsymbol{H}}}$ through the inverse formula of Stieltjes-Perron.

Lemma A.1 (Inverse Stieljies Transform)**.** *Suppose that μ is a probability measure on $\mathbb{R}$ and let m_μ be its Stieltjes transform. Then for any $a < b$, we have*

$$\mu((a,b)) + \frac{1}{2}[\mu(\{a\}) + \mu(\{b\})] = \lim_{\Im z \to 0} \frac{1}{\pi}\int_b^a \Im m_\mu(z) d\Re z$$

where $\Re$ is the function that gets the real part of z. The proof can be found at Tao [49] p. 144.

Consequently, the problem converts to obtain $\boldsymbol{G}$ if we want to obtain $\mu_{\boldsymbol{H}}$. A recent advance in the RMT community has enabled the analysis of ESD of symmetric random matrix with correlation [12] from the perspective of mean field theory [24], which we introduce in the following.

The resolvent $\boldsymbol{G}$ holds an identity by definition

$$\boldsymbol{H}\boldsymbol{G} = \boldsymbol{I} + z\boldsymbol{G} \tag{15}$$

Note that in the above equation $\boldsymbol{G}$ is a function $\boldsymbol{G}(\boldsymbol{H})$ of $\boldsymbol{H}$. When the average fluctuation of entries of $\boldsymbol{G}$ w.r.t. to its mean is small as N grows large, eq. (15) can be turned into a solvable equation regarding $\boldsymbol{G}$ instead of merely a definition. Formally, it is achieved by taking the expectation of eq. (15)

$$\mathbb{E}[\boldsymbol{H}\boldsymbol{G}] = \boldsymbol{I} + z\mathbb{E}[\boldsymbol{G}] \tag{16}$$

When fluctuation of moments beyond the second order are negligible, we can obtain a class of random matrices whose ESD can be obtained by solving a truncated cumulant expansion of eq. (16). With the above approach, using sophisticated multivariate cumulant expansion, Erdos et al. [12] proves $\boldsymbol{G}$ can be obtained as the unique solution to *Matrix Dyson Equation (MDE)* below

$$\boldsymbol{I} + (z - \boldsymbol{A} + \mathcal{S}[\boldsymbol{G}])\boldsymbol{G} = \boldsymbol{0}, \Im\boldsymbol{G} \succ \boldsymbol{0}, \Im z > 0 \tag{17}$$

where $\Im\boldsymbol{G} \succ 0$ means $\Im\boldsymbol{G}$ is positive definite,

$$\boldsymbol{A} := \mathbb{E}[\boldsymbol{H}], \frac{1}{\sqrt{N}}\boldsymbol{W} := \boldsymbol{H} - \boldsymbol{A}, \mathcal{S}[\boldsymbol{R}] := \frac{1}{N}\mathbb{E}_{\boldsymbol{W}}[\boldsymbol{W}\boldsymbol{R}\boldsymbol{W}] \tag{18}$$

$\mathcal{S}$ is a linear map on the space of $N \times N$ matrices and $\boldsymbol{W}$ is a random matrix with zero expectation. The expectation in $\mathcal{S}$ is taken w.r.t. $\boldsymbol{W}$, while taking $\boldsymbol{R}$ as a deterministic matrix.

We describe their results formally in the following, which begins with some more definitions, adopted from Erdos et al. [12].

Definition A.5 (Cumulant)**. Cumulants** *of $\kappa_{\boldsymbol{m}}$ of a random vector $\boldsymbol{w} = (w_1, \ldots, w_n)$ are defined as the coefficients of log-characteristic function*

$$\log \mathbb{E}e^{i\boldsymbol{t}^T\boldsymbol{w}} = \sum_{\boldsymbol{m}} \kappa_{\boldsymbol{m}} \frac{i\boldsymbol{t}^{\boldsymbol{m}}}{\boldsymbol{m}!}$$

where $\sum_{\boldsymbol{m}}$ is the sum over all n-dimensional multi-indices $\boldsymbol{m} = (m_1, \ldots, m_n)$.

To recall, a multi-indices is

Definition A.6 (Multi-index)**.** *a n-dimensional multi-index is an n-tuple*

$$\boldsymbol{m} = (m_1, \ldots, m_n)$$

of non-negative integers. Note that $|\boldsymbol{m}| = \sum_{i=1}^{n} m_i$, and $\boldsymbol{m}! = \prod_{i=1}^{n} m_i!$.

Similar with the more familiar concept *moment*, cumulant is also a measure of statistic properties of r.v.s. Particularly, the k-order cumulant κ characterizes the k-way correlation of a set of r.v.s. We have used the same symbols κ for covariance previously. It is because **the first order cumulant is just mean, while the second order cumulant is covariance.**

The key of insight of the paper is to identify the condition where a matrix entry $w_\alpha, \alpha \in \mathbb{I}$ is only strongly correlated with a minority of $\boldsymbol{W}_{\mathbb{I}\setminus\{\alpha\}}$, and higher order cumulants tend to be weak and not influential in large N limit. Thus, a proper formulation of the correlation strength is needed, and is defined as the cumulant norms on entries of $\boldsymbol{W}$ in the following. Given k entries $\boldsymbol{W}_{\boldsymbol{\alpha}}$ at $\boldsymbol{\alpha} = \{\alpha_i\}_{i=1,\ldots,k}, \alpha_i \in \mathbb{I}$ of matrix $\boldsymbol{W}$, where duplication is allowed, denote $\kappa(\alpha_1, \ldots, \alpha_k) = \kappa(w_{\alpha_1}, \ldots, w_{\alpha_k})$. The cumulant norms defined in [12] is summarized below. We do not want to explain the cumulant norms beyond what would be said, considering it is too technically involved and rather a distraction. **For interested readers, we suggest reading the paper [12]. In order to understand the key idea of the proof, it suffices to only understand that they characterize the strength of correlation in a certain way.** How they relate to NNs are explained in section 3, where we explain the condition in detail up to second order cumulants. Understanding higher order assumptions requires a high level of mathematical maturity given they are fresh results in the random matrix community, if readers are not familiar with the progress already.

Definition A.7 (Cumulant Norms)**.**

$$
\begin{aligned}
|||\kappa||| &:= |||\kappa|||_{\leq R} := \max_{2\leq k\leq R} |||\kappa|||_k, \\
|||\kappa|||_k &:= |||\kappa|||_k^{av} + |||\kappa|||_k^{iso} \\
|||\kappa|||_2^{av} &:= ||\,|\kappa(*,*)|\,||, \qquad \text{(19a)} \\
|||\kappa|||_k^{av} &:= N^{-2} \sum_{\alpha_1,\ldots,\alpha_k} |\kappa(\alpha_1,\ldots,\alpha_k)|, k \geq 4 \\
|||\kappa|||_3^{av} &:= ||\sum_{\alpha_1} |\kappa(\alpha_1,*,*)|\,||+ \\
&\quad \inf_{\kappa=\kappa_{dd}+\kappa_{dc}+\kappa_{cd}+\kappa_{cc}} (|||\kappa_{dd}|||_{dd} + |||\kappa_{dc}|||_{dc} \\
&\quad + |||\kappa_{cd}|||_{cd} + |||\kappa_{cc}|||_{cc}) \qquad \text{(19b)} \\
|||\kappa|||_{cc} &= |||\kappa|||_{dd} \\
&:= N^{-1} \sqrt{\sum_{b_2,a_3} (\sum_{a_2,b_3} \sum_{\alpha_1} |\kappa(\alpha_1, a_2b_2, a_3b_3)|)^2} \\
|||\kappa|||_{cd} &:= N^{-1} \sqrt{\sum_{b_3,a_1} (\sum_{a_3,b_1} \sum_{\alpha_2} |\kappa(a_1b_1, \alpha_2, a_3b_3)|)^2}, \\
|||\kappa|||_{dc} &:= N^{-1} \sqrt{\sum_{b_1,a_2} (\sum_{a_1,b_2} \sum_{\alpha_3} |\kappa(a_1b_1, a_2b_2, \alpha_3)|)^2} \\
|||\kappa|||_2^{iso} &:= \inf_{\kappa=\kappa_d+\kappa_c} (|||\kappa_d|||_d + |||\kappa_c|||_c), \\
|||\kappa|||_d &:= \sup_{||\boldsymbol{x}||\leq \mathbf{1}} ||\,||\,\kappa(\boldsymbol{x}*, \cdot *)||\,||, \\
|||\kappa|||_c &:= \sup_{||\boldsymbol{x}||\leq \mathbf{1}} ||\,||\kappa(\boldsymbol{x}*, *\cdot)||\,|| \qquad \text{(19c)} \\
|||\kappa|||_k^{iso} &:= ||\sum_{\alpha_1,\ldots,\alpha_{k-2}} |\kappa(\alpha_1,\ldots,\alpha_{k-2},*,*)|\,||, k \geq 3
\end{aligned}
$$

where in eq. (19b), the infimum is taken over all decomposition of κ in four symmetric components $\kappa_{dd}, \kappa_{dc}, \kappa_{cd}, \kappa_{cc}$; in eq. (19c) the infimum is taken over all decomposition of κ into the sum of symmetric κ_c and κ_d. The norms defined in eq. (19a) and eq. (19c) need some explanation on the notation. If in place of an index $\alpha \in \mathbb{J}$, we write a dot $(\cdot)$ in a scalar quantity then we consider the quantity as a vector indexed by the coordinate at the place of the dot. For example, $\kappa(a_1\cdot, a_2b_2)$ is a vector, the i-th entry of which is $\kappa(a_1i, a_2b_2)$ and therefore the inner norms in eq. (19c) indicate vector norms. In contrast, the outer norms indicate the operator norm of the matrix indexed by star $(*)$. More specifically, $||\boldsymbol{A}(*,*)||$ refers to the operator norm of the matrix with matrix elements $\boldsymbol{A}_{ij}$. Thus $||\,||\kappa(\boldsymbol{x}*, *\cdot)||\,||$ is the operator norm $||\boldsymbol{A}||$ of the matrix $\boldsymbol{A}$ with matrix elements $\boldsymbol{A}_{ij} = ||\kappa(\boldsymbol{x}i, j\cdot)||$. $\kappa(\boldsymbol{x}b_1, a_2b_2)$ denotes $\sum_{a_1} \kappa(a_1b_1, a_2b_2)x_{a_1}$, where $\boldsymbol{x}$ is a vector.

Before proceeding further, we expand the remark 2.8 in [12] to explain the symmetric decomposition in the above definition.

Example A.1 (Symmetry decomposition)**.** *The symmetric decomposition separates the symmetry structure in $\boldsymbol{W}$ in eq. (17), thus the cumulants arisen due to the symmetry can be tracked separately. To see an example, we look at the Wigner matrix, in which we have*

$$
\begin{aligned}
\kappa(a_1b_1, a_2b_2) &= \delta_{a_1,a_2}\delta_{b_1,b_2} + \delta_{\alpha_1,b_2}\delta_{b_1,a_2} \\
&=: \kappa_d(a_1b_1, a_2b_2) + \kappa_c(a_1b_1, a_2b_2).
\end{aligned}
$$

The cumulant naturally splits into a direct and a cross part κ_d and κ_c. Let $\boldsymbol{K}^d$ denote the matrix induced in the norm $|||\kappa_d|||$, then $\boldsymbol{K}^d_{ij} = ||\kappa_d(\boldsymbol{x}i, \cdot j)||$, which only involves the cumulant between the ith column $\boldsymbol{W}_{:,i}$ and the jth column $\boldsymbol{W}_{:,j}$. Note that due to symmetry $\boldsymbol{W}_{:,j}$ is the same as $\boldsymbol{W}_{j,:}$. Thus, to characterize/bound the cumulant between $\boldsymbol{W}_{:,i}$ and $\boldsymbol{W}_{:,j}$, the norm defined should also consider $\boldsymbol{W}_{:,j}$ as well. By decomposing κ into κ_d, κ_c, the cumulant involving $\boldsymbol{W}_{j,:}$ is taken care in $\boldsymbol{K}^c$, where *similarly, $\boldsymbol{K}^c_{ij}$ only involves with the ith column $\boldsymbol{W}_{:,i}$ and jth row $\boldsymbol{W}_{j,:}$. Thus, by decomposing κ into κ_d, κ_c, the cumulant arisen due to symmetry can be tracked separately.*

Equipped with the cumulant norms, we can state the assumptions that make MDE valid. The following boundedness and diversity assumptions are higher order correlation version of stated in asp. 3.4, 3.6.

Assumption A.1 (Bounded Mean)**.** *$\exists C \in \mathbb{R}, \forall N \in \mathbb{N}, ||\boldsymbol{A}|| \leq C$, where $||\boldsymbol{A}||$ denotes the operator norm.*

Assumption A.2 (Boundedness)**.** *1) $\forall q \in \mathbb{N}, \exists \mu_q \in \mathbb{R}, \forall \alpha \in \mathbb{I}, \mathbb{E}[|\boldsymbol{W}_\alpha|^q] \leq \mu_q$. 2) $\forall R \in \mathbb{N}, \epsilon > 0, \exists C_1, C_2(\epsilon) \in \mathbb{R}$, where $C_2(\epsilon)$ depends on ϵ, we have $|||\kappa|||_2^{iso} \leq C_1, |||\kappa||| \leq C_2(\epsilon)N^\epsilon$;*

Assumption A.3 (Diversity)**.** *There exists $\mu > 0$ such that the following holds: for every $\alpha \in \mathbb{I}$ and $q, R \in \mathbb{N}$, there exists a sequence of nested sets $\mathcal{N}_k = \mathcal{N}_k(\alpha)$ such that $\alpha \in \mathcal{N}_1 \subset \mathcal{N}_2 \subset \cdots \subset \mathcal{N}_R = \mathcal{N} \subset \mathbb{I}, |\mathcal{N}| \leq N^{1/2-\mu}$ and*

$$\kappa(f(\boldsymbol{W}_{\mathbb{I}\setminus\cup_j\mathcal{N}_{n_j+1}(\alpha_j)}), g_1(\boldsymbol{W}_{\mathcal{N}_{n_1}(\alpha_1)\setminus\cup_{j\neq 1}\mathcal{N}(\alpha_j)}), \ldots, g_q(\boldsymbol{W}_{\mathcal{N}_{n_q}(\alpha_q)\setminus\cup_{j\neq q}\mathcal{N}(\alpha_j)})) \leq N^{-3q}||f||_{q+1}\prod_{j=1}^{q}||g_j||_{q+1}$$

, for any $n_1, \ldots, n_q < R$, $\alpha_1, \ldots, \alpha_q \in \mathbb{I}$ and real analytic functions $f, g_1, \ldots, g_q$, where $||||_p$ is the L^p norm on function space. We call the set $\mathcal{N}$ of α the **coupling set** *of α.*

Remark. *In Erdos et al. [12], the functions $f, g_1, \ldots, g_q$ in asp. A.3 are assumed to be functions without any qualifiers. We change it to analytic functions for further usage. In the proof of theorem A.1, the functions are only required to be analytic, thus even if the assumptions are changed, the conclusion still holds.*

The diversity assumption requires that a matrix entry w_α only couples with a minority of the overall entries, and for the rest of the entries, the coupling strength does not exceed a certain value $N^{-3q}||f||_{q+1}\prod_{j=1}^{q}||g_j||_{q+1}$ characterized by cumulants. For example, suppose $q = 1$, given a entry α_1, the assumption essentially states that the entries in the coupling set $\mathcal{N}(\alpha_1)$ is not strongly coupled with the resting of the population $\boldsymbol{W}_{\mathbb{I}\setminus\mathcal{N}_{n_1+1}(\alpha_1)}$. The explanation goes similar as q grows, of which the coupling strengh is characterized by higher order cumulants: if we randomly pick q entries from the random matrix $\boldsymbol{H}$, the correlation of their coupling set, characterized by cumulants κ, is less than a certain value. While boundedness assumptions states the expectation of $\boldsymbol{H}$ is bounded, moments are finite, cumulants are bounded for the entries that do strongly couples. Explanation in more details in the second cumulant case can be found in section 3.4.

When the assumptions A.1 A.2 A.3 3.5 are satisfied, we have the resolvent $\boldsymbol{G}$ of a random matrix $\boldsymbol{H}$ close to the solution to the MDE probabilistically with some regularity properties as the following, adopted from Erdos et al. [12] theorem 2.2, Helton et al. [21] theorem 2.1, and Alt et al. [1] theorem 2.5.

Theorem A.1. *Assume the assumptions A.1 A.2 A.3 3.5 are satisfied for a given symmetric random matrix $\boldsymbol{H}$. Let $\boldsymbol{M}$ be the solution to the Matrix Dyson Equation eq. (17), and ρ the density function (measure) recovered from normalized trace $\frac{1}{N}tr\ \boldsymbol{M}$ through Stieljies inverse lemma A.1. We have*

a) *The MDE has a unique solution $\boldsymbol{M} = \boldsymbol{M}(z)$ for all $z \in \mathbb{H}$.*
b) *suppρ is a finite union of closed intervals with nonempty interior. Moreover, the nonempty interiors are called the* **bulk** *of the eigenvalue density function ρ.*
c) *The resolvent $\boldsymbol{G}$ of $\boldsymbol{H}$ converges to $\boldsymbol{M}$ as $N \to \infty$. The convergence rate is given by the probability bound present as follows. For any $\gamma, \epsilon > 0$, there exists $\delta > 0$, such that for all $D > 0$, give any $z \in \mathbb{D}^\delta_\gamma$, we have*

$$P(\left|\boldsymbol{x}^T(\boldsymbol{G}(z) - \boldsymbol{M}(z))\boldsymbol{y}\right| \leq ||\boldsymbol{x}||||\boldsymbol{y}||N^\epsilon/\sqrt{N\Im z}) \geq 1 - CN^{-D}$$

where $\boldsymbol{x}, \boldsymbol{y}$ are arbitrary deterministic vectors, N is the dimension of $\boldsymbol{H}$, $C > 0$ is a constant depending on D, ϵ, γ and constants in asp. A.1 A.2 A.3. ϵ can be chosen small, so $||\boldsymbol{x}||||\boldsymbol{y}||N^\epsilon/\sqrt{N\Im z}$ goes zero as N grows. The region $\mathbb{D}^\delta_\gamma$ is roughly the region in the complex plane around the intervals that are inside the support of μ. Formally, it is defined as

$$\mathbb{D}^\delta_\gamma := \{z \in \mathbb{H} \mid |z| \leq N^{C_0}, \Im z \geq N^{-1+\gamma}, \mu_{\boldsymbol{H}}(x) + dist(x, supp\mu_{\boldsymbol{H}}) \geq N^{-\delta}\},$$

where $\mathbb{H}$ is the complex domain, C_0 is a constant larger than 100, $\mu_{\boldsymbol{H}}$ is the empirical spectral distribution of $\boldsymbol{H}$ (c.f. definition A.1), and supp$(\mu_{\boldsymbol{H}})$ denotes the support of $\mu_{\boldsymbol{H}}$. Roughly, $\mathbb{D}^\delta_\gamma$ characterizes the region inside the support of $\mu_{\boldsymbol{H}}$, which is normally denoted as the bulk of the eigenspectrum.

A.3 NN loss landscape: all local minima are global minima with zero losses

We have obtained the operator equation to describe the eigen-spectrum of the Hessian of NNs and explained its assumptions. With one further assumption, we show in this section that for NNs with objective function belonging to the function class $\mathcal{L}_0$, all local minima are global minima with zero loss values.

We outline the strategy first. Since MDE is a nonlinear operator equation, it is not possible to obtain a close form analytic solution. The only way to get its solution is an iterative algorithm [21], which is not an easy task given the millions of parameters of a NN — remembering that we are dealing with large N limit — though it can serve as an exploratory tool. However, we do are able to get qualitative results by directly analyzing the equation. Our goal is to show all critical points are saddle points, except for the ones has zero loss values, which are global minima. To prove it, we prove that at the points where $R_m(T) \neq 0$, the eigen-spectrum $\mu_{\boldsymbol{H}}$ of the Hessian $\boldsymbol{H}$ is symmetric w.r.t. the y-axis, which implies that as long as non-zero eigenvalues exist, half of them will be negative. To prove it, we prove the stieltjes transform $m_{\mu_{\boldsymbol{H}}}(z)$ of $\mu_{\boldsymbol{H}}$ satisfies $\Im m_{\mu_{\boldsymbol{H}}}(-z^*) = \Im m_{\mu_{\boldsymbol{H}}}(z)$, where z^* denote the complex conjugate of z. In the following, we present the proof formally.

Lemma A.2. *Let $\boldsymbol{M}(z), \boldsymbol{M}'(-z^*)$ be the unique solution to the MDE at spectral parameter $z, -z^*$ defined at eq. (17) respectively, and $\boldsymbol{A} = \boldsymbol{0}$. We have*

$$\boldsymbol{M}' = -\boldsymbol{M}^*$$

where $*$ means taking conjugate transpose.

Proof. First, we rewrite the MDE. Note that $\mathcal{S}[\boldsymbol{G}]$ is positivity preserving, i.e., $\forall \boldsymbol{G} \succ \boldsymbol{0}, \mathcal{S}[\boldsymbol{G}] \succ \boldsymbol{0}$ by assumption 3.5. In addition, we have $\Im z > 0$, thus $\Im(z + \mathcal{S}[\boldsymbol{G}]) \succ \boldsymbol{0}$. Then, by Haagerup and Thorbjørnsen [16] lemma 3.1.(ii), we have $z + \mathcal{S}[\boldsymbol{G}]$ invertable. Thus, we can rewrite the MDE into the following form

$$\boldsymbol{G} = -(z + \mathcal{S}[\boldsymbol{G}])^{-1} \tag{20}$$

Suppose $\boldsymbol{M}$ is a solution to the MDE at spectral parameter z. The key to the proof is the fact that $\mathcal{S}[\boldsymbol{G}]$ is linear and commutes with taking conjugate, thus by replacing $\boldsymbol{M}$ with $-\boldsymbol{M}^*$, and z with $-z^*$, we would get the same equation. We show it formally in the following.

First, note that $\mathcal{S}[\boldsymbol{M}]$ is a linear map of $\boldsymbol{M}$, so the we have

$$\mathcal{S}[-\boldsymbol{M}] = -\mathcal{S}[\boldsymbol{M}]$$

Also, $\mathcal{S}[\boldsymbol{M}]$ commutes with $*$, for the fact

$$\mathcal{S}[\boldsymbol{M}^*] = \mathbb{E}[\boldsymbol{W}\boldsymbol{M}^*\boldsymbol{W}] = \mathbb{E}[(\boldsymbol{W}\boldsymbol{M}\boldsymbol{W})^*] = \mathbb{E}[\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}]^* = \mathcal{S}[\boldsymbol{M}]^*$$

Furthermore, we $*$ is commute with taking inverse, for the fact

$$\begin{aligned} & & \boldsymbol{A}\boldsymbol{A}^{-1} &= \boldsymbol{I} \\ \Longrightarrow & & (\boldsymbol{A}\boldsymbol{A}^{-1})^* &= \boldsymbol{I} \\ \Longrightarrow & & \boldsymbol{A}^{-1*}\boldsymbol{A}^* &= \boldsymbol{I} \\ \Longrightarrow & & \boldsymbol{A}^{-1*} &= \boldsymbol{A}^{*-1} \end{aligned}$$

With the commutativity results, we do the proof. The solution $\boldsymbol{M}$ satisfies the equation

$$\boldsymbol{M} = -(z + \mathcal{S}[\boldsymbol{M}])^{-1}$$

Replacing $\boldsymbol{M}$ with $-\boldsymbol{M}^*$, z with $-z^*$, we have

$$\begin{aligned} & & -\boldsymbol{M}^* &= -(-z^* + \mathcal{S}[-\boldsymbol{M}^*])^{-1} \\ \Longrightarrow & & \boldsymbol{M}^* &= -(z^* + \mathcal{S}[\boldsymbol{M}^*])^{-1} \\ \Longrightarrow & & \boldsymbol{M}^* &= -(z^* + \mathcal{S}[\boldsymbol{M}]^*)^{-1} \\ \Longrightarrow & & \boldsymbol{M}^* &= -(z + \mathcal{S}[\boldsymbol{M}])^{-1*} \\ \Longrightarrow & & \boldsymbol{M} &= -(z + \mathcal{S}[\boldsymbol{M}])^{-1} \end{aligned}$$

After the replacement, we actually get the same equation. Thus, $-\boldsymbol{M}^*, -z^*$ also satisfy eq. (20). Since the pair also satisfies the constrains $\Im \boldsymbol{M} \succ 0, \Im z > 0$, and by theorem A.1, the solution is unique, we proved the solution $\boldsymbol{M}'$ at the spectral parameter $-z^*$ is $-\boldsymbol{M}^*$.

□

Theorem A.2. *Let $\boldsymbol{H}$ be a real symmetric random matrix satisfies asp. 3.2 A.2 A.3 3.5, in addition to the assumption that $\boldsymbol{A} = \boldsymbol{0}$. Let the ESD of $\boldsymbol{H}$ be $\mu_{\boldsymbol{H}}$. Then, $\mu_{\boldsymbol{H}}$ is symmetric w.r.t. to y-axis. In other words, half of the non-zero eigenvalues are negative. Furthermore, non-zero eigenvalues always exist, implying $\boldsymbol{H}$ will always have negative eigenvalues.*

Proof. By theorem A.1, the resolvent $\boldsymbol{G}$ of $\boldsymbol{H}$ is given by the the unique solution to eq. (17) at spectral parameter z. Let the solution to eq. (17) at spectral parameter $z, -z^*$ be $\boldsymbol{M}, \boldsymbol{M}'$, By lemma A.2, we have the solutions satisfies

$$\boldsymbol{M}' = -\boldsymbol{M}^*$$

By lemma A.1, the ESD of $\boldsymbol{H}$ at $\Re z$ is given at

$$\mu_{\boldsymbol{H}}(\Re z) = \lim_{\Im z \to 0} \frac{1}{\pi} \Im m_{\mu_{\boldsymbol{H}}}(z)$$

Since $m_{\mu_{\boldsymbol{H}}}(z) = \frac{1}{N} \text{tr } M$, we have

$$\mu_{\boldsymbol{H}}(\Re z) = \lim_{\Im z \to 0} \frac{1}{\pi} \frac{1}{N} \Im \text{tr } M$$

Similarly,

$$\mu_{\boldsymbol{H}}(\Re(-z^*)) = \lim_{\Im(-z^*) \to 0} \frac{1}{\pi} \frac{1}{N} \Im \text{tr } M'$$

Note that

$$\begin{aligned}
& \mu_{\boldsymbol{H}}(\Re(-z^*)) = \lim_{\Im(-z^*) \to 0} \frac{1}{\pi} \frac{1}{N} \Im \text{tr } M' \\
\implies \quad & \mu_{\boldsymbol{H}}(\Re(-z^*)) = \lim_{\Im(-z^*) \to 0} \frac{1}{\pi} \frac{1}{N} \Im \text{tr } (-M^*) \\
\implies \quad & \mu_{\boldsymbol{H}}(-\Re z) = \lim_{\Im z \to 0} \frac{1}{\pi} \frac{1}{N} \Im \text{tr } M
\end{aligned}$$

Thus, $\mu_{\boldsymbol{H}}(\lambda), \lambda \in \mathbb{R}$ is symmetric w.r.t. y-axis. It follows that for all non-zero eigenvalues, half of them are negative.

By theorem A.1.b, there are always bulks in supp$\mu_{\boldsymbol{H}}$, thus there are always non-zero eigenvalues. Since half of the non-zero eigenvalues are negative, it follows $\boldsymbol{H}$ always has negative eigenvalues. □

Theorem (Theorem 4.1 restated)**.** *Let $R_m(T)$ be the risk function (defined at eq. (1)) of a NN T having only one output neuron (defined at eq. (3)) with a loss function l of class $\mathcal{L}_0$ (defined in section 3.1). If the Hessian $\boldsymbol{H} \in \mathbb{R}^{N \times N}$ (c.f. eq. (6)) of $R_m(T)$ satisfies assumptions 3.2 3.3 A.2 3.5 A.3, then for $R_m(T)$,*

a) all local minima are global minima with zero risk.

b) all non-zero-risk stationary points are saddle points with half of the non-zero eigenvalues negative.

c) A constant $\lambda_0 \in \mathbb{R}$ exists, such that $||\boldsymbol{H}||$ is upper bounded by $\mathbb{E}_m[l'(T(X), Y)]\lambda_0$, where $\mathbb{E}_m[l'(T(X), Y)]$ is the empirical expectation of derivative l' of l. It implies the eigen-spectrum of $\boldsymbol{H}$ is increasingly concentrated around zero as more examples have the zero loss (classified correctly in term of surrogate loss) — since when the loss $l(\boldsymbol{x}, y)$ of an example is zero, its derivative at $\boldsymbol{x}$ is zero, i.e., $l'(\boldsymbol{x}, y) = 0$ (due to the properties of $\mathcal{L}_0$). Thus, the regions around the minima are flat basins.

Remark. *A note on the generality of the theorem. First, though the network T is restricted to be of a MLP defined in eq. (3), it is chosen this way to communicate the core idea better. The results can trivially generalize to arbitrary feed forward type network architecture, e.g., Convolutional Neural Network [28], Residual Network [20].*

Remark. *It is not necessary for the assumption $\boldsymbol{A} = \boldsymbol{0}$ to be held for the theorem to hold, or more specifically, for $\boldsymbol{H}$ to have negative eigenvalues at its critical points. By proposition 2.1 in Alt et al. [1]*

$$supp\mu_{\boldsymbol{H}} \subset Spec\boldsymbol{A} + [-2||\mathcal{S}||^{1/2}, 2||\mathcal{S}||^{1/2}]$$

where Spec$\boldsymbol{A}$ denotes the support of the spectrum of the expectation matrix $\boldsymbol{A}$ and $||\mathcal{S}||$ denotes the norm induced by the operator norm. Thus, it is possible for the spectrum $\mu_{\boldsymbol{H}}$ of $\boldsymbol{H}$ to lie at the left side of the y-axis, as long as the spectrum of $\boldsymbol{A}$ is not too way off from the origin. However, existing characterizations on supp$\mu_{\boldsymbol{H}}$ based on bound are too inexact to make sure the existence of support on the left of the y-axis. To get rid of the zero expectation assumption, more works are needed to obtain a better characterization, and could be a direction for future work.

Proof of theorem 4.1. The majority of the proof has been dispersed earlier in the paper. What the proof here does mostly is to collect them into one piece.

First, we prove (a)(b) of the theorem. The Hessian $\boldsymbol{H}$ of the risk function eq. (1), can be decomposed into a summation of Hessians of loss functions of each training sample, which is described in eq. (13). For each Hessian in the decomposition, it is computed in eq. (11), and it has been shown that $\boldsymbol{H}$ is a random matrix in appendix A.1.

The analysis of the random matrix $\boldsymbol{H}$ needs to break down into two cases: 1) for all training examples, at least one example (x, y) has non-zero loss value; 2) and all training samples are classified properly with zero loss values.

We first analyze case 1), since the loss l belongs to function class $\mathcal{L}_0$, l is convex and is valued zero at its minimum. When $l(x, y) \neq 0$, we have $l'(x, y) \neq 0$, thus $\boldsymbol{H}$ is a random matrix — not a zero matrix. The analysis of this type of random matrix is undertaken in appendix A.2. For a NN satisfies asp. 3.3 A.2 A.3 3.5, and by theorem A.1, the eigen-spectrum $\mu_{\boldsymbol{H}}$ of $\boldsymbol{H}$ is given by the MDE defined at eq. (17).

By theorem A.2, $\mu_{\boldsymbol{H}}$ is symmetric w.r.t. y-axis, and half of its non-zero eigenvalues are negative. Thus, for all critical points of $R_m(T)$, its will have half of its non-zero eigenvalues negative. It implies all critical points are saddle points.

Now we turn to the case 2). In this case, all training samples are properly classified with zero loss value. Considering the lower bound of l is zero, we have reached the global minima. Also, since all critical points in case 1) are saddle points, local minima can only be reached in case 2), implying all local minima are global minima. Thus, the first part of the theorem is proved.

Now we prove (c). Note that the minima is reached for the fact that we have reached the situation where the Hessian $\boldsymbol{H}$ has degenerated into a zero matrix. Thus, each local minimum is not a traditional critical point, but a flat region, where in a local region around the minima in the parameter space, all the eigenvalues are increasingly close to zero as the parameters of the NN approach the parameters at the infimum. We show it formally in the following.

Writing a block $\boldsymbol{H}_{pq}$ (defined at eq. (10)) in the Hessian $\boldsymbol{H}_i$ of one example (defined at eq. (11)) in the form of

$$\boldsymbol{H}_{pq} = l'(T\boldsymbol{x}, y)\tilde{\boldsymbol{H}}_{pq}$$

where i is the index of the training examples, defined at eq. (1). Then, putting together $\tilde{\boldsymbol{H}}_{pq}$ together to form $\tilde{\boldsymbol{H}}_i$, $\boldsymbol{H}_i$ is rewritten in the form of

$$\boldsymbol{H}_i = l'(T\boldsymbol{x}_i, y_i)\tilde{\boldsymbol{H}}_i$$

Then the Hessian $\boldsymbol{H}$ (defined at eq. (13)) of the risk function (defined eq. (1)) can be rewritten in the form of

$$\boldsymbol{H} = \frac{1}{m}\sum_{i=1}^{m} l'(T\boldsymbol{x}_i, y_i)\tilde{\boldsymbol{H}}_i$$

Taking the operator norm on the both sides

$$||\boldsymbol{H}|| = ||\frac{1}{m}\sum_{i=1}^{m} l'(T\boldsymbol{x}_i, y_i)\tilde{\boldsymbol{H}}_i|| \leq \frac{1}{m}\sum_{i=1}^{m} |l'(T\boldsymbol{x}_i, y_i)|\,||\tilde{\boldsymbol{H}}_i||$$

Denote $\max_i\{||\tilde{\boldsymbol{H}}_i||\}$ as λ_0, we have

$$\begin{aligned}||\boldsymbol{H}|| &\leq \frac{1}{m}\sum_{i=1}^{m}|l'(T\boldsymbol{x}_i, y_i)|\lambda_0 \\ &= \mathbb{E}_m[l'(TX, Y)]\lambda_0\end{aligned}$$

The above inequality shows that, as the risk decreases, more and more samples will have zero loss value, consequently $l' = 0$, thus $\mathbb{E}_m[l']$ will be increasingly small, thus the operator norm of $\boldsymbol{H}$. At the minima where all $l' = 0$, the Hessian degenerates to a zero matrix. □

B Related works

The optimization problem of NNs has been a long standing issue for decades that attracts many attempts to understand it. Due to the sheer amount of works, we focus on the works that attack the problem in its full complexity; that is, NNs with arbitrary input data distributions, nonlinear activation functions and number of layers. We note there are emergent positive works done on shallow NNs in case readers are interested: Andoni et al. [2]; Sedghi and Anandkumar [44]; Brutzkus and Globerson [4]; Li and Yuan [31]; Zhong et al. [53]; Soltanolkotabi [46]; Haeffele and Vidal [17]; Soudry and Hoffer [48]; Venturi et al. [50]; Soltanolkotabi et al. [47]; Chizat and Bach [5]; Du and Lee [9].

Roughly, two approaches have been taken in analyzing the optimization of NNs, one from the linear algebra perspective, the other from mean field theory using random matrix theory. Our work falls in the latter approach. The linear algebra approach, as the name suggests, shies away from the nonlinear nature of the problem. Kawaguchi [25] proves all local minima of a deep linear NN are global minima when some rank conditions of the weight matrices are held. [36; 37] prove that if in a certain layer of a NN, it has more neurons than training samples, which makes it possible that the feature maps of all samples are linearly independent, then the network can reach zero training errors. A few works in the linear-algebraic direction [43; 18; 29; 32; 52; 38; 45] improve upon the two previous results, or prove similar results under different settings, but essentially use the same linear algebraic approach. As the conditions in Kawaguchi [25] indicate, the rank related linear algebraic condition does not transport to nonlinear NNs. While for Nguyen and Hein [36], it characterizes a phenomenon that if in a layer of a NN, it can allocate a neuron to memorize each training sample, then based on the memorization, it can reach zero errors. It is likely we do not need so many linearly independent intermediate features, which would lead to poor generalization. Thus, to truly understand the optimization behavior of NNs, we need to step out of the comfort zone of linearity. But we note that the linear algebraic approach is important, for finer grained analysis that may be combined with the mean field approach, in that NNs are cascaded generalized linear models.

The mean field theory approach using the tools of random matrix theory can attack the optimization problem of NNs in its full complexity, though existing works tend to be confused on the source of randomness. Due to an inadequate understanding of the randomness induced by activation function, Choromanska et al. [6] tries to get rid of the activation function by assuming that its value is independent of its input, which is unrealistic [7]. Nevertheless, it is an important attempt, and the first paper to attack deep NNs in its full complexity. After Choromanska et al. [6], which approaches by analogizing with spin glass systems — it is a complex system, as NNs are — some researchers start to study NNs from mean field theory from the first principle rather than by analog. Again, confused with the source of randomness in activation in the intermediate layers of NNs, Pennington and Bahri [40] just assumes data, weights and errors are of i.i.d. Gaussian distribution, which are mean-field assumptions and unrealistic, and proceeds to analyze the Hessian of the loss function of NNs. Due to limitations of their assumptions, they can only analyze a NN with one hidden layer.

The work by Du et al. [10] lies in a mix between the two approaches. It improves on the i.i.d. assumptions by utilizing the fact that when the number of parameter is over-parameterizingly large, a gram matrix induced by NNs is close to the random matrix at initialization, thus it enables NNs to converge to zero risk. However, the convergence requires the training data are linearly independent, as done in the works in the linear algebraic approach. As stated, it again sidesteps the randomness in NNs and relies on random matrix at initialization, similarly as having been done by [7; 40], while we directly analyze the Hessian matrix at any time during training. As one of the consequences, we obtain results on NNs, e.g., LeNet, that are not necessarily over-parameterized, while in [10], it requires the width of each layer to grow quartically w.r.t. the number of training samples.

The above approaches work in the parameter space, Jacot et al. [23] approaches from the dual space by studying a kernel named Neural Tangent Kernel. It shows that if the width of each layer of a NN goes to infinite literally, the training reaches global minima. The proof relies heavily on the delicate Gaussian structure in the last layer created by the infinity, and does not easily generalize to cases where the infinity does not hold.

Lastly, since all works that give results on deep nonlinear neural networks rely on assumptions that the network size should be large, we compare the requirements of different works. In our results, the dimension N of Hessian is only required to be sufficiently large, instead of literally being infinity. This is a non-asymptotic result that is different from existing results: N does not depend on the training set size as in [36; 37; 10], and thus N does not need to grow polynomially or exponentially with them; the result holds non-asymptotically, instead of asymptotically as in Neural Tangent Kernel (NTK) [23], whose results cannot hold if layer width $n \not\to \infty$. For NNs, N is normally very large. In the toy example we discussed in fig. 2, N is 10^8, which makes $\boldsymbol{H}$ a *very* large random matrix. The experiment in section 1 shows that the N of a relatively small LeNet is large enough.

C Assumption simplification in Gaussian ensembles

We explain why asp. 3.4 3.6 present in section 3.4 are simplified the version of asp. A.2 A.3 present in appendix A.3.

Recall that the objective function is the empirical risk function $R_m(T)$ at eq. (1). Given a set of i.i.d. training samples $(X_i, Y_i)_{i=1,\ldots,m}$, $R_m(T)$ is a summation of the i.i.d. random matrices. Formally, reusing the notation to denote $\boldsymbol{H}$ the Hessian of $R_m(T)$ and $\boldsymbol{H}_i$ the Hessian of $l(T(X_i; \boldsymbol{\theta}), Y_i)$, we have

$$\boldsymbol{H} = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{H}_i \tag{21}$$

Acute readers may realize that by the multivariate central limiting theorem (Klenke [27] theorem 15.57), $\boldsymbol{H}$ will converge to a Gaussian ensemble, i.e., a random matrix of which the distribution of entries is a Gaussian process (GP), asymptotically as $m \to \infty$. This is why we need asp. 3.1.

When $\boldsymbol{H}$ is a Gaussian ensemble, all higher order cumulants vanishes, thus the diversity assumption is solely about the second order cumulants, and the case when $q = 1$. So are higher cumulants in the boundedness assumption. Thus, it suffices to have asp. 3.4 3.6 only involve cumulants up to second order. As mentioned in appendix A.2, the first order cumulant is mean, while the second order cumulant is covariance. In asp. 3.6, we only need the $q = 1$ is due to the how multivariate cumulant expansion is used in the proof in [12], which is rather involved and cannot be explained without delving into the proof. We suggest the interested readers to read the paper [12], since it is not possible to explain a close-to-100-page proof here. Since f, $\{g_i\}_{i=1,\ldots,q}$ are analytic, $\kappa(f(\cdot), g_1(\cdot))$ is a generalized cumulant (McCullagh [35] Chapter 3), which can be decomposed into a sum of cumulants of entries of the Hessian. Considering that only first and second cumulants exist, which are means and covariance respectively, $\kappa(f(\cdot), g_1(\cdot))$ *is thus a sum of means and covariance of the Hessian's entries*. Thus, it characterizes the weakness of the entries that are not correlated discussed previously in section 3.4.

D Experiment details

We describe the details of the experiments we have shown in section 1. The eigen-spectrum is computed for a classical LeNet type NN [30] on the MNIST dataset with Lanczos spectrum approximation algorithms [33; 39; 13]. All experiments are run with PyTorch [41].

Model & Training. No simplification is done on the LeNet type NN. We only replace the cross entropy loss with hinge loss, so that the network can be used for binary classification. The NN has architecture as in table 1. The NN is trained with Stochastic Gradient Descent with momentum 0.9. The NN is trained for 10 epochs. The initial learning rate is 0.01, and decayed at 6th epoch to 0.001. The network is initialized with a very old initialization method that initializes each weight by sampling from the uniform distribution with standard deviation $\frac{1}{\sqrt{n}}$, where n is the number of input channels. No explicit regularization is used, e.g., weight decay [28].

NN Architecture
Conv $5 \times 5 \times 32$
ReLU
MaxPool 2×2
Conv $5 \times 5 \times 64$
ReLU
MaxPool 2×2
Fully Connected 512
ReLU
Fully Connected 1
Hinge Loss

Table 1: LeNet Network Architecture. On the right of "Conv" is kernel height × kernel width × the output channel number. On the right of "MaxPool" is the kernel size. On the right of "FullyConnected" is output channel number. Each convolution layer uses strides 1, while pooling layer uses strides the same as the kernel size.

Dataset. We train the NN on the classic MNIST data. Since we need to do binary classification, we take label 0 as the positive class, and the rest as the negative class. To create a balanced dataset, we use all images that are digital 0, and an equal number of samples of the negative class, which are randomly sampled digital from $1 - 9$. The training set consists of about 12000 samples, and the validation set consists of about 2000 samples. The samples are preprocessed to be zero-centered and of unit variance. No data augmentation is used.

Lanczos Spectrum Approximation Algorithm. We use the Lanczos approximation algorithm in Papyan [39], which is proposed by Lin et al. [33], and implemented contemporarily by Ghorbani et al. [13]. The eigen-spectum shown in fig. 1b is computed on 200 training samples. Similar to what has been done in Papyan [39], when computing the extreme eigenvalues to normalize the eigen-spectrum to $[-1, 1]$, we run 128 iterations with $\kappa = 0.05$ (the iteration number M_0 and margin percentage κ in `Normalization` algorithm, i.e., Algorithm 4, in [39] respectively); when approximating the eigen-spectrum of Hessian, i.e., the `LanczosApproxSpec` algorithm (Algorithm 3), we use iteration number $M = 128$, the number of points $K = 1024$, and the number of starting random vector $n_{\text{vec}} = 1$.