Unraveling the Hessian: A Key to Smooth Convergence in Loss Function Landscapes

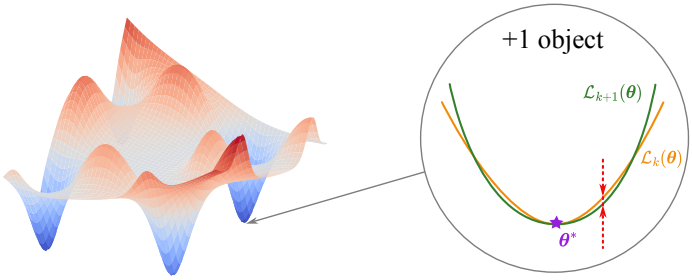
Nikita Kiselev
MIPT, Sber AI
Moscow, Russia
kiselev.ns@phystech.edu

Andrey Grabovoy
MIPT
Moscow, Russia
grabovoy.av@phystech.edu

Abstract

The loss landscape of neural networks is a critical aspect of their training, and understanding its properties is essential for improving their performance. In this paper, we investigate how the loss surface changes when the sample size increases, a previously unexplored issue. We theoretically analyze the convergence of the loss landscape in a fully connected neural network and derive upper bounds for the difference in loss function values when adding a new object to the sample. Our empirical study confirms these results on various datasets, demonstrating the convergence of the loss function surface for image classification tasks. Our findings provide insights into the local geometry of neural loss landscapes and have implications for the development of sample size determination techniques.

1 Introduction



(a) Loss function landscape (b) Losses difference

Figure 1: **Overview of our observations.** Part (a) shows the loss function landscape, which is a surface in the parameters space. Part (b) shows the losses difference. It arises, when one more object is added to the dataset. Here we exhibit the behavior for dimension equals 2. Near the minimum $\boldsymbol{\theta}^*$, the mean loss value for $k+1$ objects $\mathcal{L}_{k+1}(\boldsymbol{\theta})$ tends to be similar to the same for k objects $\mathcal{L}_k(\boldsymbol{\theta})$.

The advancement of neural networks has significantly driven the development of optimization methods [1–3]. Nevertheless, a notable challenge lies in the extensive number of parameters, which can result in numerous potential local and global minima of the loss function [4–7]. Many studies have endeavored to identify the most optimal minima from diverse perspectives [8–10].

Flatter minima are associated with superior generalization properties. This issue has been explored theoretically and empirically in many studies, including [11–14]. The process of locating a flat minimum is not straightforward. In such contexts, the Hessian matrix, which represents the second-order derivative of the loss function, is frequently utilized. This matrix is crucial for understanding the local behavior of the function around the point of extremum.

Preprint. Under review.

Existing research [15–22] has addressed the question of Hessian spectra in typical neural network architectures, revealing a characteristic pattern. The spectrum typically comprises a significant number of eigenvalues close to zero, referred to as bulk eigenvalues, and a smaller number of distinct eigenvalues, known as outliers [15]. Specifically, in a K-label classification task, precisely K non-zero eigenvalues can be observed [16, 17]. Theoretical estimations of these non-vanishing eigenvalues enable the establishment of an upper bound on the local rate of growth of the loss function.

However, the issue of altering the landscape of the loss function when adding new objects to the sample remains unresolved. In this paper, we meticulously investigate how the loss surface transforms as the sample size increases. Specifically, we examine the absolute value of the loss function changes when adding another object. An overview of our observations is presented in Figure 1.

We obtain theoretical estimates for the convergence of this difference in a fully connected neural network as the sample size approaches infinity. These results are derived through the analysis of the Hessian spectrum. These estimates allow us to determine the dependence of this difference on the structure of the neural network, including the size of the hidden layer and the number of layers. We empirically verify these theoretical results by examining the behavior of the loss surface on various datasets. The obtained plots substantiate the validity of the theoretical calculations.

Contributions. Our contributions can be summarized as follows:

- We present a theoretical analysis of the convergence of the loss landscape in a fully connected neural network as the sample size increases, deriving upper bounds for the difference in loss function values when adding a new object to the sample.
- We demonstrate the validity of our theoretical results through empirical studies on various datasets, showing that the loss function surface exhibits convergence for image classification tasks.
- We highlight the implications of our findings for understanding the local geometry of neural loss landscapes and for the development of sample size determination techniques, addressing a previously unexplored issue in the field.

Outline. The rest of the paper is organized as follows. Section 2 divides existing results into some topics, highlighting their key contributions and findings. Section 3 considers general notation and some preliminary calculations. In Section 4, we provide theoretical bounds on the hessian and losses difference norms. Empirical study of the obtained results is given in Section 5. We summarize and present the results in Sections 6 and 7. Additional experiments and proofs of theorems are included in the Appendix A.

2 Related Work

Understanding the Geometry of Neural Network Loss Landscapes. The geometry of neural network loss landscapes, particularly through the Hessian matrix, has been extensively studied. [14] identifies key properties including that in the multi-label classificaion problem the landscape exhibits exactly K directions of high positive curvature, where K is the number of classes. [23] and [15] use random matrix theory and spectral analysis to understand the loss surface dynamics and optimization. [24] introduces a model for understanding linear mode connectivity (LMC) by analyzing the topography of the loss landscape. [25] provides a theoretical understanding of the double descent phenomenon in finite-width neural networks, leveraging influence functions to derive expressions for the population loss. [26] characterizes the instabilities of gradient descent during training with large learning rates, observing landscape flattening and shift. However, nowhere is the question raised that this geometry becomes unchanged with an increase in the number of objects.

Hessian-Based Generalization and Optimization. The Hessian matrix is crucial for studying generalization and optimization in neural networks. In [27] a Hessian-based distance for fine-tuning against label noise was introduced, which can match the scale of the observed generalization gap of fine-tuned models in practice. [28] optimizes for wider local minima using training data, while concurrently maintaining low loss values on validation data to improve generalization. Authors of [29] ties loss curvature to input-output behavior, explaining progressive sharpening and flat minima. All these studies provide insights into optimizing and generalizing neural networks using Hessian-based methods. However, none of these papers address the impact of changing sample size.

Spectral Analysis and Structural Insights. Spectral analysis of the Hessian matrix offers insights into neural network structure and properties. There is a variety of works [15–17] underlying that the Hessian matrix of typical loss surface exhibits a spectrum composed of two parts: a bulk centered near zero and outliers away from the bulk. [18] shows that depending on the data properties, the nonlinear response model, and the loss function, the Hessian can have qualitatively different spectral behaviors. [19] highlights the Hessian's role in sharpness regularization. [20] reveals class/cross-class structure in Hessian spectra. [17] and [21] analyze Hessian dynamics and hierarchical structure. [22] unifies low-dimensional observations, explaining Hessian spectra and gradient descent alignment. [30] and [16] analyze the Hessian's eigenvalue distribution, highlighting over-parametrization and data dependency. [31] discovers and mathematically models the power-law Hessian spectrum, providing a maximum entropy interpretation and a framework for spectral analysis in deep learning. However, these low-rank approximations have not been sufficiently investigated in terms of the convergence of the corresponding spectra.

Decomposing and Analyzing the Hessian Matrix. Decomposing the Hessian matrix provides insights into neural network training and generalization. [32] proposes a decoupling conjecture to analyze Hessian properties, decomposing it as the Kronecker product of two smaller matrices. [33] explores CNN Hessian maps, revealing the Hessian rank grows as the square root of the number of parameters. [34] provides exact formulas and tight upper bounds for the Hessian rank of deep linear networks. [35] simplifies high-dimensional derivative calculations using tensor calculus. However, the limit properties of the Hessian with increasing sample size have not been sufficiently investigated.

3 Preliminaries

3.1 General notation

In this section, we introduce the general notation used in the rest of the paper and the basic assumptions. Consider a conditional probability $p(\mathbf{y}|\mathbf{x})$, that maps the given unobserved variable $\mathbf{x} \in \mathcal{X}$ to the corresponding output $\mathbf{y} \in \mathcal{Y}$, and which we try to approximate using neural network $f_{\boldsymbol{\theta}}(\cdot)$ with parameters $\boldsymbol{\theta} \in \mathbb{R}^P$.

Given the dataset

$$\mathfrak{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}, \quad i = 1, \ldots, m,$$

the empirical loss function calculated for all the given dataset of size m is

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{m} \sum_{i=1}^{m} \ell(f_{\boldsymbol{\theta}}(\mathbf{x}_i), \mathbf{y}_i) \approx \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p(\mathbf{x},\mathbf{y})} \left[\ell(f_{\boldsymbol{\theta}}(\mathbf{x}), \mathbf{y})\right],$$

so it is an approximation of general loss function, which is calculated using the joint distribution $p(\mathbf{x}, \mathbf{y})$. Here we denote the per-object loss function as $\ell(\mathbf{z}, \mathbf{y})$, e.g., cross-entropy loss $\mathrm{CE}(\mathrm{softmax}(\mathbf{z}), \mathbf{y})$ in multi-label classification. If we fix first k samples, corresponding loss function is

$$\mathcal{L}_k(\boldsymbol{\theta}) = \frac{1}{k} \sum_{i=1}^{k} \ell(f_{\boldsymbol{\theta}}(\mathbf{x}_i), \mathbf{y}_i).$$

Difference between losses for sample sizes $k+1$ and k is

$$\mathcal{L}_{k+1}(\boldsymbol{\theta}) - \mathcal{L}_k(\boldsymbol{\theta}) = \frac{1}{k+1} \left(\ell(f_{\boldsymbol{\theta}}(\mathbf{x}_{k+1}), \mathbf{y}_{k+1}) - \mathcal{L}_k(\boldsymbol{\theta})\right). \tag{1}$$

The further investigation in the work is aimed at studying exactly this difference, which occurs when adding another object to the dataset. We are especially interested in limiting properties when the sample size tends to infinity.

Assumption 1. *Let $\boldsymbol{\theta}^*$ be the local minimum of both empirical loss functions $\mathcal{L}_k(\boldsymbol{\theta})$ and $\mathcal{L}_{k+1}(\boldsymbol{\theta})$, i.e. $\nabla\mathcal{L}_k(\boldsymbol{\theta}^*) = \nabla\mathcal{L}_{k+1}(\boldsymbol{\theta}^*) = \mathbf{0}$.*

This assumption will allow us to investigate the behavior of the loss function landscape, limiting ourselves to considering just one point.

3.2 Second-order approximation

Let us use second-order Taylor approximation for mentioned above loss functions at $\boldsymbol{\theta}^*$. We suppose that decomposition to the second order will be sufficient to study local behavior. The first-order term vanishes because the gradients $\nabla\mathcal{L}_k(\boldsymbol{\theta}^*)$ and $\nabla\mathcal{L}_{k+1}(\boldsymbol{\theta}^*)$ zeroes:

$$\mathcal{L}_k(\boldsymbol{\theta}) \approx \mathcal{L}_k(\boldsymbol{\theta}^*) + \frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\theta}^*)^\intercal \mathbf{H}^{(k)}(\boldsymbol{\theta}^*)(\boldsymbol{\theta}-\boldsymbol{\theta}^*), \tag{2}$$

where we denoted the Hessian of $\mathcal{L}_k(\boldsymbol{\theta})$ w.r.t. parameters $\boldsymbol{\theta}$ at $\boldsymbol{\theta}^*$ as $\mathbf{H}^{(k)}(\boldsymbol{\theta}^*) \in \mathbb{R}^{P\times P}$. Moreover, the total Hessian can be written as the average value of the Hessians of the individual terms of the empirical loss function:

$$\mathbf{H}^{(k)}(\boldsymbol{\theta}) = \nabla^2_{\boldsymbol{\theta}}\mathcal{L}_k(\boldsymbol{\theta}) = \frac{1}{k}\sum_{i=1}^{k}\nabla^2_{\boldsymbol{\theta}}\ell(f_{\boldsymbol{\theta}}(\mathbf{x}_i),\mathbf{y}_i) = \frac{1}{k}\sum_{i=1}^{k}\mathbf{H}_i(\boldsymbol{\theta}).$$

Therefore, using the obtained second-order approximation (2), formula for the difference of losses (1) becomes

$$\mathcal{L}_{k+1}(\boldsymbol{\theta}) - \mathcal{L}_k(\boldsymbol{\theta}) = \frac{1}{k+1}\left(\ell(f_{\boldsymbol{\theta}^*}(\mathbf{x}_{k+1}),\mathbf{y}_{k+1}) - \frac{1}{k}\sum_{i=1}^{k}\ell(f_{\boldsymbol{\theta}^*}(\mathbf{x}_i),\mathbf{y}_i)\right) +$$

$$+\frac{1}{k+1}(\boldsymbol{\theta}-\boldsymbol{\theta}^*)^\intercal\left(\mathbf{H}_{k+1}(\boldsymbol{\theta}^*) - \frac{1}{k}\sum_{i=1}^{k}\mathbf{H}_i(\boldsymbol{\theta}^*)\right)(\boldsymbol{\theta}-\boldsymbol{\theta}^*).$$

After that, using triangle inequality, we can derive the following:

$$|\mathcal{L}_{k+1}(\boldsymbol{\theta}) - \mathcal{L}_k(\boldsymbol{\theta})| \leqslant \frac{1}{k+1}\left|\ell(f_{\boldsymbol{\theta}^*}(\mathbf{x}_{k+1}),\mathbf{y}_{k+1}) - \frac{1}{k}\sum_{i=1}^{k}\ell(f_{\boldsymbol{\theta}^*}(\mathbf{x}_i),\mathbf{y}_i)\right| +$$

$$+\frac{1}{k+1}\left\|\boldsymbol{\theta}-\boldsymbol{\theta}^*\right\|_2^2\left\|\mathbf{H}_{k+1}(\boldsymbol{\theta}^*) - \frac{1}{k}\sum_{i=1}^{k}\mathbf{H}_i(\boldsymbol{\theta}^*)\right\|_2.$$

So the problem of the boundedness and convergence of the losses difference is reduced to the analysis of the two terms:

- Difference of the **loss functions at optima** for new object and previous ones:

$$\left|\ell(f_{\boldsymbol{\theta}^*}(\mathbf{x}_{k+1}),\mathbf{y}_{k+1}) - \frac{1}{k}\sum_{i=1}^{k}\ell(f_{\boldsymbol{\theta}^*}(\mathbf{x}_i),\mathbf{y}_i)\right|,$$

- Difference of the **Hessians at optima** for new object and previous ones:

$$\left\|\mathbf{H}_{k+1}(\boldsymbol{\theta}^*) - \frac{1}{k}\sum_{i=1}^{k}\mathbf{H}_i(\boldsymbol{\theta}^*)\right\|_2.$$

It should be mentioned that the first term can be easily upper-bounded by a constant, since the loss function itself takes limited values. However, the expression with Hessians is not so easy to evaluate. The rest of the work is devoted to a thorough analysis of this difference. Thus, we analyze the local convergence of the landscape of the loss function using its Hessian.

3.3 Fully connected neural network

The main results of this work are obtained for K-label classification problem with a cross-entropy loss function. In this case the input vector is $\mathbf{x} \in \mathbb{R}^n$ and the output $\mathbf{y} \in \mathbb{R}^K$, which is a one-hot vector, with all components equal to 0 except a single component y_k if and only if k is the correct class label for input $\mathbf{x}$. Consider a L-layer fully connected network $f_{\boldsymbol{\theta}}(\cdot)$ with ReLU activation

function after each linear layer. With $\sigma(\mathbf{x}) = [\mathbf{x} \geqslant \mathbf{0}]\,\mathbf{x}$ as the Rectified Linear Unit (ReLU) function, the output of this network is a vector of logits $\mathbf{z} \in \mathbb{R}^K$. They are computed recursively as

$$\mathbf{z}^{(p)} = \mathbf{W}^{(p)}\mathbf{x}^{(p)} + \mathbf{b}^{(p)},$$
$$\mathbf{x}^{(p+1)} = \sigma(\mathbf{z}^{(p)}).$$

Here we denote the input and output of the p-th layer as $\mathbf{x}^{(p)}$ and $\mathbf{z}^{(p)}$, and set $\mathbf{x}^{(1)} = \mathbf{x}$, $\mathbf{z} = f_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{z}^{(L)}$. Also we denote $\boldsymbol{\theta} = \mathrm{col}(\mathbf{w}^{(1)}, \mathbf{b}^{(1)}, \ldots, \mathbf{w}^{(L)}, \mathbf{b}^{(L)}) \in \mathbb{R}^P$ the parameters of the network. For the p-th layer, $\mathbf{w}^{(p)}$ is the flattened weight matrix $\mathbf{W}^{(p)}$ and $\mathbf{b}^{(p)}$ is its corresponding bias vector. We denote $\mathbf{p} = \mathrm{softmax}(\mathbf{z}) \in \mathbb{R}^K$ as the output confidence, i.e.

$$p_i = \mathrm{softmax}(\mathbf{z})_i = \frac{\exp(z_i)}{\sum_{j=1}^K \exp(z_j)} \in (0; 1).$$

The loss function is cross-entropy loss:

$$\ell(\mathbf{z}, \mathbf{y}) = \mathrm{CE}(\mathbf{p}, \mathbf{y}) = -\sum_{k=1}^K y_k \log p_k \in \mathbb{R}^+.$$

3.4 Hessian decomposition

It is quite well known [16] that, via the chain rule [35], the Hessian can be decomposed as a sum of the following two matrices:

$$\mathbf{H}_i(\boldsymbol{\theta}) = \underbrace{\nabla_{\boldsymbol{\theta}}\mathbf{z}_i \frac{\partial^2 \ell(\mathbf{z}_i, \mathbf{y}_i)}{\partial \mathbf{z}_i^2} \nabla_{\boldsymbol{\theta}}\mathbf{z}_i^\intercal}_{\text{G-term}} + \underbrace{\sum_{k=1}^K \frac{\partial \ell(\mathbf{z}_i, \mathbf{y}_i)}{\partial z_{ik}} \nabla^2_{\boldsymbol{\theta}} z_{ik}}_{\text{H-term}},$$

where $\nabla_{\boldsymbol{\theta}}\mathbf{z}_i \in \mathbb{R}^{P \times K}$ is the Jacobian of the neural network function and $\dfrac{\partial^2 \ell(\mathbf{z}_i, \mathbf{y}_i)}{\partial \mathbf{z}_i^2}$ is the Hessian of the loss with respect to the network function, at the i-th sample. As it was mentioned [15–17], the Hessian spectrum consists of bulk which is concentrated around zero (corresponds to the H-term), and the edges which are scattered away from zero (G-term). We will focus on the top eigenspace, so we can approximate our full Hessian using only G-term, as

$$\mathbf{H}_i(\boldsymbol{\theta}) \approx \nabla_{\boldsymbol{\theta}}\mathbf{z}_i \frac{\partial^2 \ell(\mathbf{z}_i, \mathbf{y}_i)}{\partial \mathbf{z}_i^2} \nabla_{\boldsymbol{\theta}}\mathbf{z}_i^\intercal.$$

Moreover, recent works exploring the Neural Tangent Kernel [36, 37] assume that the logits $\mathbf{z}$ depend only linearly on the weights $\boldsymbol{\theta}$, which implies that the logit curvatures $\nabla^2_{\boldsymbol{\theta}} z_{ik}$, and therefore the H-term are identically zero.

4 Convergence of the loss difference

In this section, we present the results obtained regarding the bounding the Hessian norm. After that, we apply the obtained upper bound on the Hessian to get a rate of the losses difference convergence. Despite the fact that this is only an upper bound and it is not always achieved, in the Section 5 we analyze the general trends for practical application.

Authors of [38] derived the formula of G-term in the fully connected neural network, so in our case we use the following approximation: $\mathbf{H}_i(\boldsymbol{\theta}) \approx \mathbf{F}_i^\intercal \mathbf{A}_i \mathbf{F}_i$. Here the following is denoted (we omit the index i for simplicity):

- Matrix representation of the ReLU activation function:

$$\mathbf{D}^{(p)} = \mathrm{diag}([\mathbf{z}^{(p)} \geqslant \mathbf{0}]),$$

- The partial derivative of logits w.r.t. logits at p-th layer:

$$\mathbf{G}^{(p)} = \frac{\partial \mathbf{z}}{\partial \mathbf{z}^{(p)}} = \mathbf{W}^{(L)}\mathbf{D}^{(L-1)}\mathbf{W}^{(L-1)}\mathbf{D}^{(L-2)} \cdot \ldots \cdot \mathbf{D}^{(p)},$$

- Its stacked version:

$$\mathbf{F}^\intercal = \begin{pmatrix} (\mathbf{G}^{(1)})^\intercal \otimes \mathbf{x}^{(1)} \\ (\mathbf{G}^{(1)})^\intercal \\ \vdots \\ (\mathbf{G}^{(L)})^\intercal \otimes \mathbf{x}^{(L)} \\ (\mathbf{G}^{(L)})^\intercal \end{pmatrix},$$

- And the Hessian of the loss function w.r.t. logits (according to [39]):

$$\mathbf{A} = \nabla^2_{\mathbf{z}} \ell(\mathbf{z}, \mathbf{y}) = \operatorname{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}^\intercal.$$

To the best of our knowledge, no one had received an estimate for the norms of such an approximation of the Hessian previously. Next, we present a Theorem 1 that contains an upper bound on the spectral norm of the Hessian in a fully connected neural network.

4.1 Boundedness of the Hessian

Below there is a Theorem 1, a detailed proof of which is provided in Appendix A.2.

Theorem 1. *Consider a L-layer fully connected neural network with ReLU activation function and without bias terms, applied to solve a K-label classification problem. Suppose the following is satisfied: $\|\mathbf{W}^{(p)}\|_2 \leqslant M_{\mathbf{W}}$ and $\|\mathbf{x}_i\|_2 \leqslant M_{\mathbf{x}}$ for all layers $p = 1, \ldots, L$ in network and for all objects $i = 1, \ldots, m$ in the dataset. Then, for any object $i = 1, \ldots, m$ the following inequality holds:*

$$\|\mathbf{H}_i(\boldsymbol{\theta})\|_2 \leqslant L\sqrt{2} M_{\mathbf{x}}^2 M_{\mathbf{W}}^{2L} + \sqrt{2}\frac{M_{\mathbf{W}}^2(M_{\mathbf{W}}^{2L} - 1)}{M_{\mathbf{W}}^2 - 1}.$$

This theorem allows us to understand the dependence of the Hessian norm on the structure of the neural network: the size of the hidden layer and the number of layers. So, the next Lemma 1 exhibits the dependence on the hidden size.

Lemma 1. *If each model parameter is bounded by a constant $M > 0$, that is $|w_{ij}^{(p)}| \leqslant M$ for all $i, j = 1, \ldots, h$ and for all layers $p = 1, \ldots, L$, then, under the conditions of Theorem 1, the following is true:*

$$\|\mathbf{H}_i(\boldsymbol{\theta})\|_2 \leqslant L\sqrt{2} M_{\mathbf{x}}^2 (hM)^{2L} + \sqrt{2}\frac{(hM)^2((hM)^{2L} - 1)}{(hM)^2 - 1}.$$

So, the following proportionality holds:

$$\|\mathbf{H}_i(\boldsymbol{\theta})\|_2 \propto L(hM)^{2L}.$$

Obtained results allows us to claim that the Hessian norm is a power function of the size of the hidden layer h and an exponential function of the number of layers L. Although it may seem that the estimate received is too high, this is actually not the case. The fact is that if we choose h to be large, then the limiting constant M will most likely be very small. Because of this, the number under the power of $2L$ will probably be less than one. Further, we use this upper bound to get an inequality for the loss function difference.

4.2 Losses difference convergence

Below there is a Lemma 2, a detailed proof of which is provided in Appendix A.4

Lemma 2. *Let $\boldsymbol{\theta}$ be chosen as $\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_2^2 \leqslant R^2$ for some $R > 0$. If there exist a non-negative constant M_ℓ such $|\ell(f_{\boldsymbol{\theta}^*}(\mathbf{x}_i), \mathbf{y}_i)| \leqslant M_\ell$ for all objects $i = 1, \ldots, m$ in the dataset, then, under the conditions of Theorem 1, the following holds:*

$$|\mathcal{L}_{k+1}(\boldsymbol{\theta}) - \mathcal{L}_k(\boldsymbol{\theta})| \leqslant \frac{2}{k+1}\left(M_\ell + \left(L\sqrt{2} M_{\mathbf{x}}^2 M_{\mathbf{W}}^{2L} + \sqrt{2}\frac{M_{\mathbf{W}}^2(M_{\mathbf{W}}^{2L} - 1)}{M_{\mathbf{W}}^2 - 1}\right) R^2\right) \to 0 \text{ as } k \to \infty.$$

So, the following proportionality is true:

$$|\mathcal{L}_{k+1}(\boldsymbol{\theta}) - \mathcal{L}_k(\boldsymbol{\theta})| \propto \frac{L(hM)^{2L} R^2}{k}. \tag{3}$$

Based on the estimates obtained, the following conclusions can be drawn. Firstly, the convergence of the loss function surface is affected by the distance from the extremum point. The farther we are from this point, the slower the convergence may be. Secondly, the situation regarding the number of layers and layer size is similar to that of the Hessian matrix. An increase in the number of layers L will lead to a higher convergence score, while an increase in layer size h does not necessarily have a negative impact. Again, this is due to the constant that evaluates the magnitude of each element in the weight matrix. Finally, the resulting estimate is inversely proportional to the sample size, exhibiting a sublinear rate of convergence. In the next section, we demonstrate the typical behavior of the loss function surface in practice and compare it with the theoretical one obtained.

5 Experiments

To verify the theoretical estimates obtained, we conducted a detailed empirical study. In this section, we present the results from training a fully connected neural network for the **Image Classification** task. The experiments can be easily reproduced by following the instructions provided in our GitHub repository: https://github.com/kisnikser/landscape-hessian.

The primary objective of these experiments is to empirically confirm the convergence of the loss landscape as the sample size increases. To achieve this, we trained a fully connected neural network on **the entire dataset** and obtained the corresponding parameters $\hat{\boldsymbol{\theta}}$ as a point near the minimum. Subsequently, we examined the relationship between the average loss difference and the available sample size.

We utilized the `pytorch` library [40] as the Python framework for neural network training. The architecture employed is consistent with that described in Section 3, consisting of several linear layers with a ReLU activation function after each layer, except the final one. The size h was fixed for all hidden layers L. The network was trained over numerous epochs using the `Adam` optimizer [41] with a constant learning rate of 10^{-3}. Various Image Classification datasets available in the `torchvision` library were used. While the graphs presented below are based on a single dataset, additional results for each dataset can be found in Appendix A.1. For the training process, a batch size of 64 was selected. The experiments were conducted on a Tesla A100 80GB GPU with 16 CPU cores and 243 GB of RAM.

5.1 Direct Image Classification

In this experiment, we utilized the pixel values of images as inputs. Figure 2 displays the results obtained from an analysis of 10,000 objects from the MNIST dataset [42], with the network trained over 10 epochs. The corresponding input size is 784, while the output size is 10. The plots on the left were generated by fixing the number of layers at $L = 5$ in the network. The hidden size across all layers was varied from 4 to 64. Concurrently, the figure on the right illustrates the behavior of the loss difference as the number of hidden layers changes from 1 to 10, with the hidden size h fixed at 16. This sequence was repeated 100 times for averaging. An exponential moving average with a smoothing factor of 0.99 was applied to the obtained results.

From the dependencies observed, it is evident that although the change is not substantial, adding more layers leads to a greater difference in the loss functions (see the **right** side). Conversely, increasing the hidden size results in a smaller difference between the loss functions (see the **left** side).

For readers unfamiliar with the topic, this may be surprising, as the estimation we have made (3) suggests a power dependence on the layer size h, and the value of L is clearly not less than 1. However, readers are referred to Section 4.2 for a more detailed discussion of this phenomenon. Additionally, in practice, the constant M that limits the magnitude of weights is found to be relatively small. Furthermore, since the MNIST dataset classification task is considered relatively straightforward, a shallow yet wide neural network can produce good classification results. Consequently, the values of the loss function have been observed to be lower for larger h values, and therefore their difference is also lower.

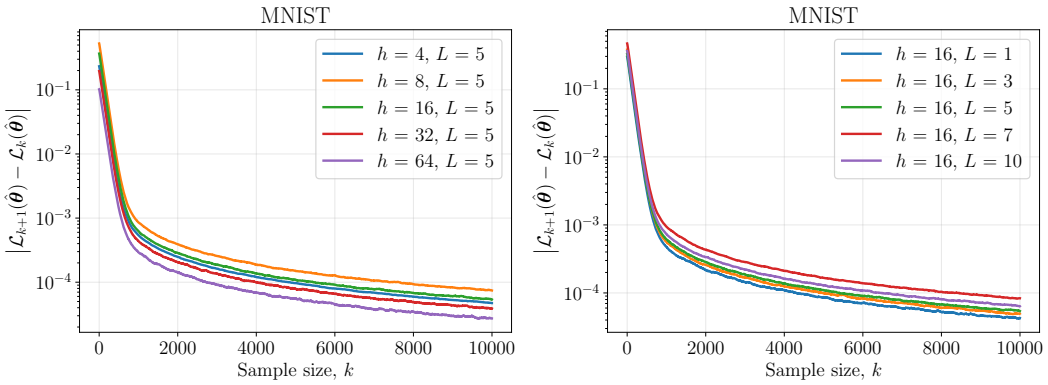


Figure 2: The dependence of the absolute value of the loss function difference on the available sample size, **direct image classification**. The graphs on the left show a decrease in values as the dimension of the hidden layer increases. The graphs on the right show an increase in values as the number of layers increases.

5.2 Image features extraction

In contrast to the previous experiment, this part employs a pre-trained image feature extractor. The fully connected network is utilized as a multi-label classification head. We selected the Vision Transformer (ViT) [43] from Google.

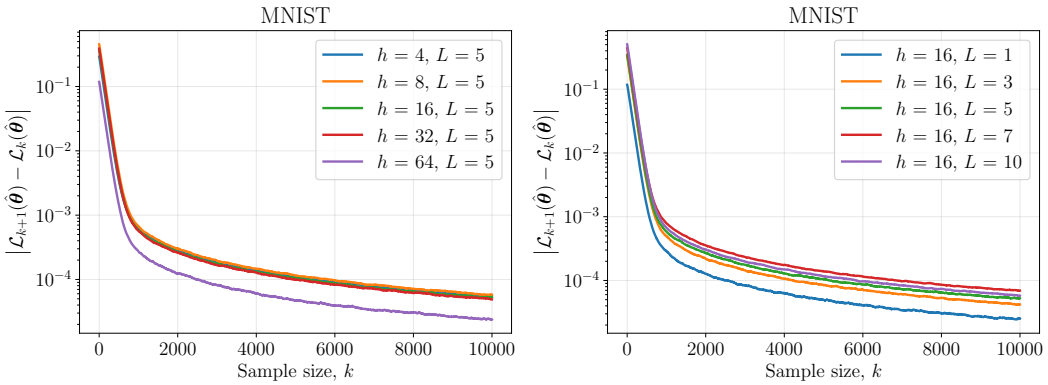


Figure 3: The dependence of the absolute value of the loss function difference on the available sample size, **image features extraction**. The graphs on the left show a decrease in values as the dimension of the hidden layer increases. The graphs on the right show an increase in values as the number of layers increases.

We similarly selected 10,000 objects randomly from the MNIST dataset and varied the hidden size and the number of layers. The results are consistent with those observed in direct image classification. This consistency confirms that the convergence presented does not depend on the nature of the space of the original objects $\mathcal{X}$. Boundedness of this space is sufficient to observe the convergence of the loss function landscape.

Our experiment corroborates the convergence proved in Lemma 2. Additionally, the upper bound on the rate of this convergence holds true. Indeed, altering the parameters of the neural network, such as the number of layers and the layer size, leads to a slight change in the difference of the loss functions. We remind the reader that a larger number of graphs can be found in Appendix A.1.

6 Discussion

The results of this study provide insights into the loss landscape convergence as the dataset size increases. Our theoretical analysis shows that the absolute difference between the average loss function values when adding one more object to the sample tends to zero, as the number of available objects tends to infinity. This was achieved by proving the upper-bound theorem for the Hessian norm in a fully connected neural network. Empirical study allows us to confirm our results practically. In particular, we claim that the loss function surface exhibits convergence for the Image Classification task, both as a direct classifier of initial representations and as a multi-label classification head after the pre-trained feature extractor.

The results of our research are highly connected to the problem of the local geometry of neural loss landscapes. Despite the fact that a large number of studies have been devoted to this issue, the change in dataset size has remained a significant gap. In this paper, we have tried to take the first steps in this direction.

Nevertheless, this study has potential limitations. First, our theoretical analysis was deterministic and not probabilistic. Although this may bring certain clarifications to the estimates, we do not think it can have a serious impact in practice. Second, the subject of our research is a fully connected neural network. We plan to extend our results to other architectures in future work. Third, using Assumption 1, we suppose existing such a point, which will be a minimum, starting with a certain sample size. Finally, using the triangle inequality in the proof of Lemma 2 yields a rough upper bound for the loss difference, so it may be improved in future work.

We believe that our findings will contribute to the development of more precise studies of the behavior of the loss landscape when changing the training sample size. We also believe that our results can be used to develop modern sample-size determination techniques. We expect this because the convergence of the loss function landscape can be considered as the sign that the training sample size is sufficient for the selected model. In future work, we hope to apply our observations to this field.

7 Conclusion

In this paper, we have presented a comprehensive study of the convergence of the loss landscape in a fully connected neural network as the sample size increases. Our theoretical analysis and empirical results demonstrate that the absolute difference between the average loss function values when adding one more object to the sample tends to zero as the number of available objects tends to infinity. These findings provide valuable insights into the local geometry of neural loss landscapes and address a previously unexplored issue in the field. We believe that our results will contribute to the development of more precise studies of the loss landscape behavior and have implications for the development of sample size determination techniques. Future work will focus on extending our results to other architectures and improving the upper bounds for the loss difference.

References

[1] Dami Choi, Christopher J. Shallue, Zachary Nado, Jaehoon Lee, Chris J. Maddison, and George E. Dahl. On empirical comparisons of optimizers for deep learning, 2020. URL https://arxiv.org/abs/1910.05446.

[2] Derya Soydaner. A comparison of optimization algorithms for deep learning. *International Journal of Pattern Recognition and Artificial Intelligence*, 34(13):2052013, April 2020. ISSN 1793-6381. doi: 10.1142/s0218001420520138. URL http://dx.doi.org/10.1142/S0218001420520138.

[3] Robin M. Schmidt, Frank Schneider, and Philipp Hennig. Descending through a crowded valley - benchmarking deep learning optimizers, 2021. URL https://arxiv.org/abs/2007.01547.

[4] Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. The role of over-parametrization in generalization of neural networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=BygfghAcYX.

[5] Difan Zou and Quanquan Gu. An improved analysis of training over-parameterized deep neural networks, 2019. URL https://arxiv.org/abs/1906.04688.

[6] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization, 2019. URL https://arxiv.org/abs/1811.03962.

[7] Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and generalization in overparameterized neural networks, going beyond two layers, 2020. URL https://arxiv.org/abs/1811.04918.

[8] Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks, 2015. URL https://arxiv.org/abs/1412.0233.

[9] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks, 2017. URL https://arxiv.org/abs/1708.03888.

[10] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets, 2018. URL https://arxiv.org/abs/1712.09913.

[11] Sepp Hochreiter and Jürgen Schmidhuber. Simplifying neural nets by discovering flat minima. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7. MIT Press, 1994.

[12] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. Exploring generalization in deep learning, 2017. URL https://arxiv.org/abs/1706.08947.

[13] Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets, 2017. URL https://arxiv.org/abs/1703.04933.

[14] Stanislav Fort and Surya Ganguli. Emergent properties of the local geometry of neural loss landscapes, 2019. URL https://arxiv.org/abs/1910.05929.

[15] Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An investigation into neural net optimization via hessian eigenvalue density. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2232–2241. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/ghorbani19b.html.

[16] Levent Sagun, Utku Evci, V. Ugur Guney, Yann Dauphin, and Leon Bottou. Empirical analysis of the hessian of over-parametrized neural networks, 2018. URL https://arxiv.org/abs/1706.04454.

[17] Vardan Papyan. The full spectrum of deepnet hessians at scale: Dynamics with sgd training and sample size, 2019. URL https://arxiv.org/abs/1811.07062.

[18] Zhenyu Liao and Michael W. Mahoney. Hessian eigenspectra of more realistic nonlinear models, 2021. URL https://arxiv.org/abs/2103.01519.

[19] Yann N. Dauphin, Atish Agarwala, and Hossein Mobahi. Neglected hessian component explains mysteries in sharpness regularization, 2024. URL https://arxiv.org/abs/2401.10809.

[20] Vardan Papyan. Traces of class/cross-class structure pervade deep learning spectra, 2020. URL https://arxiv.org/abs/2008.11865.

[21] Vardan Papyan. Measurements of three-level hierarchical structure in the outliers in the spectrum of deepnet hessians, 2019. URL https://arxiv.org/abs/1901.08244.

[22] Connall Garrod and Jonathan P. Keating. Unifying low dimensional observations in deep learning through the deep linear unconstrained feature model, 2024. URL https://arxiv.org/abs/2404.06106.

[23] Jeffrey Pennington and Yasaman Bahri. Geometry of neural network loss surfaces via random matrix theory. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2798–2806. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/pennington17a.html.

[24] Sidak Pal Singh, Linara Adilova, Michael Kamp, Asja Fischer, Bernhard Schölkopf, and Thomas Hofmann. Landscaping linear mode connectivity, 2024. URL https://arxiv.org/abs/2406.16300.

[25] Sidak Pal Singh, Aurelien Lucchi, Thomas Hofmann, and Bernhard Schölkopf. Phenomenology of double descent in finite-width neural networks, 2022. URL https://arxiv.org/abs/2203.07337.

[26] Lawrence Wang and Stephen Roberts. The instabilities of large learning rate training: a loss landscape view, 2023. URL https://arxiv.org/abs/2307.11948.

[27] Haotian Ju, Dongyue Li, and Hongyang R. Zhang. Robust fine-tuning of deep neural networks with hessian-based generalization guarantees, 2023. URL https://arxiv.org/abs/2206.02659.

[28] Van-Anh Nguyen, Quyen Tran, Tuan Truong, Thanh-Toan Do, Dinh Phung, and Trung Le. Agnostic sharpness-aware minimization, 2024. URL https://arxiv.org/abs/2406.07107.

[29] Lachlan Ewen MacDonald, Jack Valmadre, and Simon Lucey. On progressive sharpening, flat minima and generalisation, 2023. URL https://arxiv.org/abs/2305.14683.

[30] Levent Sagun, Leon Bottou, and Yann LeCun. Eigenvalues of the hessian in deep learning: Singularity and beyond, 2017. URL https://arxiv.org/abs/1611.07476.

[31] Zeke Xie, Qian-Yuan Tang, Yunfeng Cai, Mingming Sun, and Ping Li. On the power-law hessian spectrums in deep learning, 2022. URL https://arxiv.org/abs/2201.13011.

[32] Yikai Wu, Xingyu Zhu, Chenwei Wu, Annie Wang, and Rong Ge. Dissecting hessian: Understanding common structure of hessian in neural networks, 2022. URL https://arxiv.org/abs/2010.04261.

[33] Sidak Pal Singh, Thomas Hofmann, and Bernhard Schölkopf. The hessian perspective into the nature of convolutional neural networks, 2023. URL https://arxiv.org/abs/2305.09088.

[34] Sidak Pal Singh, Gregor Bachmann, and Thomas Hofmann. Analytic insights into structure and rank of neural network hessian maps, 2021. URL https://arxiv.org/abs/2106.16225.

[35] Maciej Skorski. Chain rules for hessian and higher derivatives made easy by tensor calculus, 2019. URL https://arxiv.org/abs/1911.13292.

[36] Arthur Jacot, Franck Gabriel, and Clement Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[37] Jaehoon Lee, Lechao Xiao, Samuel S Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent*. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(12):124002, dec 2020. doi: 10.1088/1742-5468/abc62b. URL https://dx.doi.org/10.1088/1742-5468/abc62b.

[38] Yikai Wu, Xingyu Zhu, Chenwei Wu, Annie Wang, and Rong Ge. Dissecting hessian: Understanding common structure of hessian in neural networks, 2022.

[39] Sahil Singla, Eric Wallace, Shi Feng, and Soheil Feizi. Understanding impacts of high-order loss approximations and features in deep learning interpretation, 2019.

[40] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[41] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA, 2015.

[42] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[43] Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Zhicheng Yan, Masayoshi Tomizuka, Joseph Gonzalez, Kurt Keutzer, and Peter Vajda. Visual transformers: Token-based image representation and processing for computer vision, 2020.

[44] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017. URL `https://arxiv.org/abs/1708.07747`.

[45] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. URL `https://api.semanticscholar.org/CorpusID:18268744`.

A Appendix / supplemental material

A.1 Additional experiments

In this section, we provide an extended version of the conducted experiments. As it was discussed in the Section 5, we trained a fully connected neural network for the Image Classification task. Below there is a Table 1 with a description of the datasets used. We have chosen four datasets from the `torchivison` library: MNIST [42], FashionMNIST [44], CIFAR10 and CIFAR100 [45]. The only preprocessing of the data is normalization to bring the values into the range $[-1; 1]$.

Table 1: Image Classification datasets description

Name	Description	Format	Resolution
MNIST [42]	Handwritten digits	Grayscale	28×28
FashionMNIST [44]	Fashion clothing items	Grayscale	28×28
CIFAR10 [45]	Various objects	RGB	32×32
CIFAR100 [45]	Various objects	RGB	32×32

We have already discussed the results for the MNIST dataset in Section 5, so the following plots are only for the remained sets.

Direct image classification. Here we used pixel values of images as inputs. Plots on the left were gained when fixing the number of layers $L = 5$ in the network. We changed the hidden size on the all layers from 4 to 64. At the same time, the figure on the right demonstrates the behavior of the loss difference when the number of hidden layers changes from 1 to 10, but the size $h = 16$ remains unchanged. This sequence was repeated 100 times for averaging. For the obtained results, we applied an exponential moving average with smoothing factor 0.99.

Image features extraction. Unlike the previous experiment part, here we use a pre-trained image feature extractor firstly.

Similarly to Section 5, the results confirm the convergence proved in the Lemma 2. Also, the upper bound on the rate of this convergence is true. Specifically, changing the parameters of the neural network: the number of layers and the size of the layer leads to the change in the difference of the loss functions.

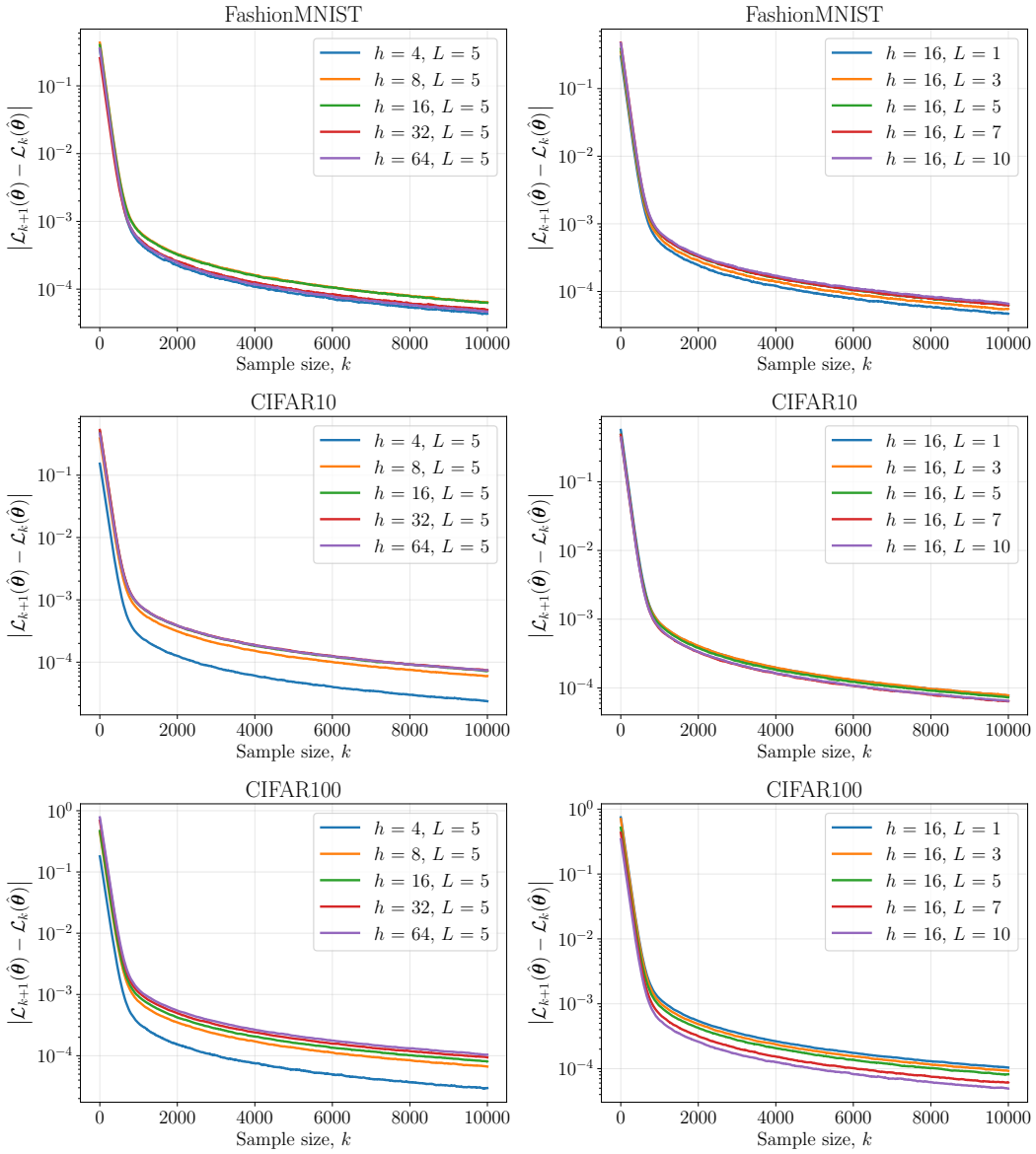


Figure 4: The dependence of the absolute value of the loss function difference on the available sample size, **direct image classification**. The graphs on the left show a decrease in values as the dimension of the hidden layer increases. The graphs on the right show an increase in values as the number of layers increases. Results on different datasets: FashionMNIST, CIFAR10 and CIFAR100.

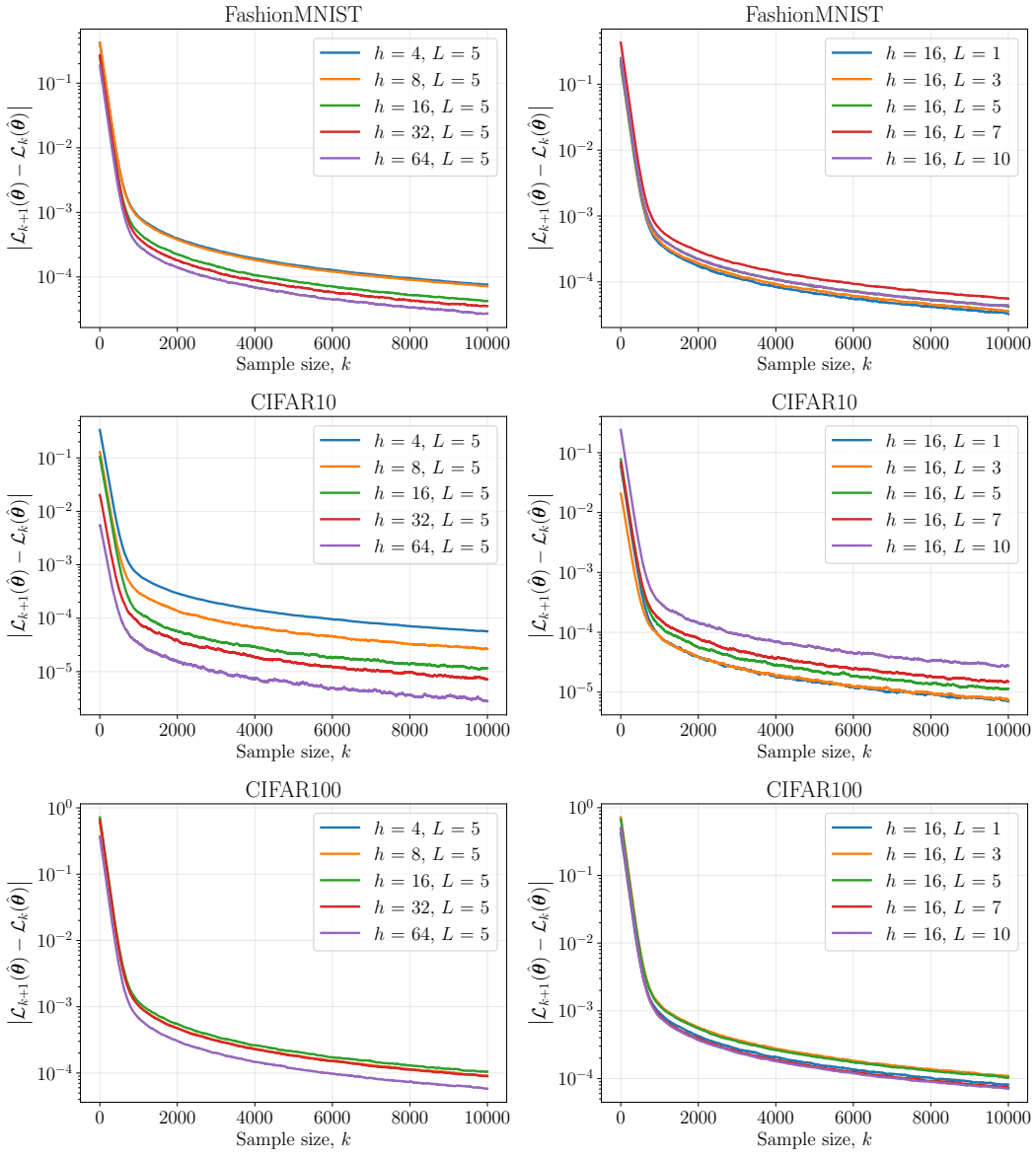


Figure 5: The dependence of the absolute value of the loss function difference on the available sample size, **image features extraction**. The graphs on the left show a decrease in values as the dimension of the hidden layer increases. The graphs on the right show an increase in values as the number of layers increases. Results on different datasets: FashionMNIST, CIFAR10 and CIFAR100.

A.2 Proof of Theorem 1

Proof. For simplicity, we omit the index i, which corresponds to the particular object in dataset. Firstly, because the spectral matrix norm is sub-multiplicative, we get

$$\|\mathbf{H}(\boldsymbol{\theta})\|_2 = \left\|\mathbf{F}^\intercal \mathbf{A} \mathbf{F}\right\|_2 \leqslant \|\mathbf{A}\|_2 \|\mathbf{F}\|_2^2 .$$

Next, we will focus on considering each term separately. $\|\mathbf{A}\|_2$: Due to the norm equivalence, the following inequality holds:

$$\|\mathbf{A}\|_2 \leqslant \|\mathbf{A}\|_F ,$$

where $\|\mathbf{A}\|_F$ is the Frobenius norm. So, further we will use some key properties of this norm to estimate the above term. Using the definition of Frobenius norm,

$$\|\mathbf{A}\|_F^2 = \left\|\mathrm{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}^\intercal\right\|_F^2 = \sum_{k=1}^{K}(p_k - p_k^2)^2 + \sum_{k\neq l} p_k^2 p_l^2 = \sum_{k=1}^{K} p_k^2(1-p_k)^2 + \sum_{k\neq l} p_k^2 p_l^2.$$

Since $0 \leqslant p_k \leqslant 1$ for all $i = 1, \ldots, K$, we have $0 \leqslant p_k^2 \leqslant p_k$ and $0 \leqslant (1-p_k)^2 \leqslant (1-p_k)$. Consequently, we can derive the following inequalities:

$$\sum_{k=1}^{K} p_k^2(1-p_k)^2 \leqslant \sum_{k=1}^{K} p_k(1-p_k) \leqslant \sum_{k=1}^{K} p_k = 1,$$

$$\sum_{k\neq l} p_k^2 p_l^2 \leqslant \sum_{k\neq l} p_k p_l = \left(\sum_{k=1}^{K} p_k\right)^2 - \sum_{k=1}^{K} p_k^2 \leqslant 1 - \sum_{k=1}^{K} p_k^2.$$

Combining these inequalities, we obtain:

$$\|\mathbf{A}\|_F^2 \leqslant 1 + 1 - \sum_{k=1}^{K} p_k^2 = 2 - \sum_{k=1}^{K} p_k^2 \leqslant 2,$$

thus, the Frobenius norm of $\mathbf{A}$ is upper-bounded by $\sqrt{2}$, therefore

$$\|\mathbf{A}\|_2 \leqslant \|\mathbf{A}\|_F \leqslant \sqrt{2}.$$

$\|\mathbf{F}\|_2$: To obtain a bound on the $\|\mathbf{F}\|_2$, we firstly analyze the spectral norm of the matrix

$$\mathbf{G}^{(p)} = \mathbf{W}^{(L)}\mathbf{D}^{(L-1)}\mathbf{W}^{(L-1)}\mathbf{D}^{(L-2)} \cdot \ldots \cdot \mathbf{D}^{(p)}.$$

Using the sub-multiplicative property of the spectral norm, we have:

$$\|\mathbf{G}^{(p)}\|_2 \leqslant \|\mathbf{W}^{(L)}\|_2 \|\mathbf{D}^{(L-1)}\|_2 \|\mathbf{W}^{(L-1)}\|_2 \|\mathbf{D}^{(L-2)}\|_2 \cdot \ldots \cdot \|\mathbf{D}^{(p)}\|_2.$$

Since $\mathbf{D}^{(p)}$ is a diagonal matrix with entries equal to 0 or 1, its spectral norm is upper-bounded by 1. Therefore, we can simplify the above inequality as:

$$\|\mathbf{G}^{(p)}\|_2 \leqslant \prod_{s=p}^{L} \|\mathbf{W}^{(s)}\|_2.$$

Then, using the property that squared spectral norm of vertically-stacked matrices is less or equal to the sum of their squared spectral norms (it is easy to observe), we get:

$$\|\mathbf{F}\|_2^2 \leqslant \sum_{p=1}^{L} \left(\|(\mathbf{G}^{(1)})^\intercal \otimes \mathbf{x}^{(1)}\|_2^2 + \|(\mathbf{G}^{(1)})^\intercal\|_2^2\right).$$

Spectral norm of the Kronecker matrix product is equal to their ordinary product norm, i.e.

$$\|\mathbf{F}\|_2^2 \leqslant \sum_{p=1}^{L} \|\mathbf{G}^{(p)}\|_2^2 \left(\|\mathbf{x}^{(p)}\|_2^2 + 1\right).$$

Further, we substitute the upper-bound obtained above and have:

$$\|\mathbf{F}\|_2^2 \leqslant \sum_{p=1}^{L} \left(\|\mathbf{x}^{(p)}\|_2^2 + 1\right) \prod_{s=p}^{L} \|\mathbf{W}^{(s)}\|_2^2.$$

So the final bound we get for the Hessian is (we substitute the object index again):

$$\|\mathbf{H}_i(\boldsymbol{\theta})\|_2 \leqslant \sqrt{2} \sum_{p=1}^{L} \left(\|\mathbf{x}_i^{(p)}\|_2^2 + 1\right) \prod_{s=p}^{L} \|\mathbf{W}^{(s)}\|_2^2.$$

To the simplicity, we will omit bias terms, i.e. set $\mathbf{b}^{(p)} = \mathbf{0}$ for all $p = 1, \ldots, L$, then we get the following:

$$\|\mathbf{x}_i^{(p)}\|_2 \leqslant \|\mathbf{x}_i\|_2 \prod_{s=1}^{p-1} \|\mathbf{W}^{(s)}\|_2,$$

and therefore

$$\|\mathbf{H}_i(\boldsymbol{\theta})\|_2 \leqslant \sqrt{2} \sum_{p=1}^{L} \left(\|\mathbf{x}_i\|_2^2 \prod_{s=1}^{p-1} \|\mathbf{W}^{(s)}\|_2^2 + 1 \right) \prod_{s=p}^{L} \|\mathbf{W}^{(s)}\|_2^2 =$$

$$= L\sqrt{2} \|\mathbf{x}_i\|_2^2 \prod_{p=1}^{L} \|\mathbf{W}^{(p)}\|_2^2 + \sqrt{2} \sum_{p=1}^{L} \prod_{s=p}^{L} \|\mathbf{W}^{(s)}\|_2^2.$$

If $\|\mathbf{W}^{(p)}\|_2 \leqslant M_{\mathbf{W}}$ and $\|\mathbf{x}_i\|_2 \leqslant M_{\mathbf{x}}$, then

$$\|\mathbf{H}_i(\boldsymbol{\theta})\|_2 \leqslant L\sqrt{2} M_{\mathbf{x}}^2 M_{\mathbf{W}}^{2L} + \sqrt{2} \frac{M_{\mathbf{W}}^2 (M_{\mathbf{W}}^{2L} - 1)}{M_{\mathbf{W}}^2 - 1}.$$

A separate interesting case is when $L = 1$, then

$$\|\mathbf{H}_i(\boldsymbol{\theta})\|_2 \leqslant \sqrt{2} M_{\mathbf{W}}^2 \left(M_{\mathbf{x}}^2 + 1\right).$$

□

A.3 Proof of Lemma 1

Proof. The relationship between the spectral norm and the Frobenius norm, as well as the definition of the Frobenius norm, allow us to obtain

$$\|\mathbf{W}^{(p)}\|_2^2 \leqslant \|\mathbf{W}^{(p)}\|_F^2 = \sum_{i,j=1}^{h} \left(w_{ij}^{(p)}\right)^2 \leqslant h^2 M^2,$$

where M is such a constant, that $\left(w_{ij}^{(p)}\right)^2 \leqslant M^2$ for all $i, j = 1, \ldots, h$ and for all $p = 1, \ldots, L$.

$$h^2 M^2 \leqslant M_{\mathbf{W}}^2,$$

therefore

$$\|\mathbf{H}_i(\boldsymbol{\theta})\|_2 \leqslant L\sqrt{2} M_{\mathbf{x}}^2 (hM)^{2L} + \sqrt{2} \frac{(hM)^2((hM)^{2L} - 1)}{(hM)^2 - 1}.$$

So, the following proportionality is true:

$$\|\mathbf{H}_i(\boldsymbol{\theta})\|_2 \propto L h^{2L}.$$

A separate interesting case is when $L = 1$, then

$$\|\mathbf{H}_i(\boldsymbol{\theta})\|_2 \leqslant \sqrt{2} M_{\mathbf{W}}^2 \left(M_{\mathbf{x}}^2 + 1\right),$$

that is

$$\|\mathbf{H}_i(\boldsymbol{\theta})\|_2 \propto h^2.$$

□

A.4 Proof of Lemma 2

Proof. Using the triangle inequality for the mentioned above terms, we get

$$\left|\ell(f_{\boldsymbol{\theta}^*}(\mathbf{x}_{k+1}), \mathbf{y}_{k+1}) - \frac{1}{k}\sum_{i=1}^{k}\ell(f_{\boldsymbol{\theta}^*}(\mathbf{x}_i), \mathbf{y}_i)\right| \leqslant$$

$$\leqslant |\ell(f_{\boldsymbol{\theta}^*}(\mathbf{x}_{k+1}), \mathbf{y}_{k+1})| + \left|\frac{1}{k}\sum_{i=1}^{k}\ell(f_{\boldsymbol{\theta}^*}(\mathbf{x}_i), \mathbf{y}_i)\right| \leqslant$$

$$\leqslant |\ell(f_{\boldsymbol{\theta}^*}(\mathbf{x}_{k+1}), \mathbf{y}_{k+1})| + \frac{1}{k}\sum_{i=1}^{k}|\ell(f_{\boldsymbol{\theta}^*}(\mathbf{x}_i), \mathbf{y}_i)| \leqslant$$

Then, due to the boundedness of loss on train samples, we obtain

$$\leqslant M_\ell + \frac{1}{k}\sum_{i=1}^{k} M_\ell = 2M_\ell = \mathcal{O}(1) \text{ as } k \to \infty.$$

Similarly for Hessians, it is easy to get

$$\left\|\mathbf{H}_{k+1}(\boldsymbol{\theta}^*) - \frac{1}{k}\sum_{i=1}^{k}\mathbf{H}_i(\boldsymbol{\theta}^*)\right\|_2 \leqslant$$

$$\leqslant \|\mathbf{H}_{k+1}(\boldsymbol{\theta}^*)\| + \left\|\frac{1}{k}\sum_{i=1}^{k}\mathbf{H}_i(\boldsymbol{\theta}^*)\right\|_2 \leqslant$$

$$\leqslant \|\mathbf{H}_{k+1}(\boldsymbol{\theta}^*)\| + \frac{1}{k}\sum_{i=1}^{k}\|\mathbf{H}_i(\boldsymbol{\theta}^*)\|_2 \leqslant$$

$$\leqslant M_{\mathbf{H}} + \frac{1}{k}\sum_{i=1}^{k} M_{\mathbf{H}} = 2M_{\mathbf{H}} = \mathcal{O}(1) \text{ as } k \to \infty,$$

where from Theorem 1 we get

$$M_{\mathbf{H}} = L\sqrt{2}M_{\mathbf{x}}^2 M_{\mathbf{W}}^{2L} + \sqrt{2}\frac{M_{\mathbf{W}}^2(M_{\mathbf{W}}^{2L} - 1)}{M_{\mathbf{W}}^2 - 1}.$$

Thus, substituting the obtained estimates into the expression for the difference, we receive

$$|\mathcal{L}_{k+1}(\boldsymbol{\theta}) - \mathcal{L}_k(\boldsymbol{\theta})| \leqslant \frac{2M_\ell}{k+1} + \frac{2M_{\mathbf{H}}}{k+1}\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_2^2.$$

Choosing the particular neighborhood of the local minimum $\boldsymbol{\theta}^*$, i.e. $\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_2^2 \leqslant R^2$, we get

$$|\mathcal{L}_{k+1}(\boldsymbol{\theta}) - \mathcal{L}_k(\boldsymbol{\theta})| \leqslant \frac{2}{k+1}\left(M_\ell + M_{\mathbf{H}}R^2\right) \to 0 \text{ as } k \to \infty.$$

□