ConvNets Landscape Convergence: Hessian-Based Analysis of Matricized Networks

Abstract—**The Hessian of a neural network is an important aspect for understanding the loss landscape and the characteristic of network architecture. The Hessian matrix captures important information about the curvature, sensitivity, and local behavior of the loss function. Our work proposes a method that enhances the understanding of the local behavior of the loss function and can be used to analyze the behavior of neural networks and also for interpreting the parameters in these networks. In this paper, we consider an approach to investigate the properties of the deep neural network, using the Hessian. We propose a method for estimating the Hessian matrix norm for a specific type of neural networks like convolutional. We have obtained the results for both 1D and 2D convolutions, as well as for the fully connected head in these networks. Our empirical analysis supports these findings, demonstrating convergence in the loss function landscape. We have evaluated the Hessian norm for neural networks represented as a product of matrices and considered how this estimate affects the landscape of the loss function.**

I. Introduction

The loss landscape is important for understanding the nature of deep network parameters [1]–[3]. Often it helps to understand differences between architectures [4], [5], significance of activation [6] and local and global minima properties [7]–[9]. Many studies have investigated the loss function landscape for modern architectures, for instance, [10] investigates the self-supervised ViT through the lens of the loss landscape, [11] investigates ViTs and MLP-Mixers from the lens of loss geometry, intending to improve the models' data efficiency at training and generalization at inference. Paper [12] presents a landscape visualization method that provides useful insights about neural network loss landscapes. The authors of [13] reduce the time required to compute such loss landscapes. Studies [14], [15] conduct experimental exploration on the loss surface of the deep neural network, including trajectories of various adaptive optimization algorithms. Numerous works have examined the spectrum of the Hessian matrix [16]–[18] or established upper bounds on the rank of a matrix. For instance, [19] estimated the ranks of Hessian blocks through their decomposition.

In this work, we derive theoretical estimates for the spectral norm of the Hessian. We demonstrate that the spectral norm of the Hessian provides an upper bound on the difference between the mean values of the loss function when incorporating an additional object into the sample. This leads to further applications related to sample size determination [20] and offers insights into the manner in which parameters can influence the Hessian matrix. We propose a theoretical analysis for decomposing the Hessian into linear components and provide specific applications to convolutional networks. First, we investigate the behavior of the loss function in the vicinity of the optimum. We examine how the loss landscape near the solution and the Hessian norm depend on the specific network architecture.

Another objective of this study is to understand how the norm of parameters, along with their quantity and spatial distribution within the neural network, influences the learning process [21]–[23], where the meaning of parameters and ways to reduce their number are described in detail. Additionally, a crucial aspect of this work is the evaluation of the absolute difference between the average loss function values at successive steps, which is derived directly from our estimation of the Hessian norm.

Our contributions can be summarized as follows:

- We present a method for the decomposition of the Hessian matrix into linear components and apply this approach to estimate the norm of the Hessian matrix.
- We illustrate the application of our results to convolutional architectures and provide insights into the relationship between parameters and their corresponding estimates.
- We demonstrate the validity of our theoretical results through experiments on image classification, using convolutional networks.

II. Related Work

Neural Network Loss Landscapes. The landscape of loss functions has been explored from various perspectives in the literature. For instance, [24] show connections between a number of classes and directions of high positive curvature. Paper [5] shown, that the loss landscape of the two-layer ReLU network has good properties when the number of hidden nodes is large. Work [25] offers a model of how the loss landscape needs to behave topographically for LMC(linear mode connectivity). Work [26] categorizes the loss surfaces curves, plotted along Gaussian noise directions. The properties of neural networks and their Hessian spectra near the interpolation threshold are uncovered by [27]. Works [10], [11] explore the ViT architecture using the local landscape of the loss function. However, these works are based on specific architectures.

Analyzing the Hessian Matrix. The decomposition of the Hessian matrix into its components is a key tool for studying its properties. The decoupling conjecture, which decomposes the layer-wise Hessians of a network as the Kronecker product of two smaller matrices, is proposed by [28]. In [29] we can see Hessian chain rule and useful tensor calculations associated with it. However, these studies have not been extended to analyze the loss function landscape through the use of the Hessian matrix norm.

Hessian eigenvalues and eigenspectra. The spectrum of the Hessian matrix is crucial for understanding the structure of the loss function landscape. Authors of the work [16] develop a tool to study the evolution of the entire Hessian spectrum throughout the optimization process. The authors of [30] efficiently approximate the spectrum of the Hessian of neural networks, through decomposing the Hessian into different components. At work [31] we can see empirical evidence, that eigenvalue distribution can be composed of two parts: which is concentrated around zero, and which are scattered away from zero. Authors of [32] identify and discuss an important formal class/cross-class structure and show how it lies at the origin of the many features observed in deep networks spectra. The existence of outliers in the spectrum of the Hessian is addressed by [33], who attempt to provide an explanation. The authors of [18] make a characterization of the Hessian eigenspectra for a broad family of nonlinear models. Paper [21] develops a tool to study the evolution of the entire Hessian spectrum throughout the optimization process. The authors of [34] have covered the topic most accurately, the paper proposes a method for investigating the spectrum when changing the sample size, but they are limited to only the fully connected neural network.

III. Preliminaries

A. General notation

In this section, we introduce the general notation used in the rest of the paper and the basic assumptions. Similar to [35] we consider matrix derivatives, using vectorize row-wise (vec_r). For given $\mathbf{X} \in \mathbb{R}^{m\times n}, \mathbf{Y} \in \mathbb{R}^{p\times q}$ we define:

$$\frac{\partial \mathbf{X}}{\partial \mathbf{Y}} := \frac{\partial vec_r \mathbf{X}}{\partial (vec_r \mathbf{Y})^\intercal}.$$

For higher tensor dimension we define it in the same way. Having K-label classification problem, we consider a probability $p(\mathbf{y}|\mathbf{x})$, that maps unobserved $\mathbf{x} \in \mathcal{X}$ to corresponding output $\mathbf{y} \in \mathcal{Y} = \mathbb{R}^K$ — one-hot vectors, where K — number of classes. We have a neural network $f_{\boldsymbol{\theta}}$, parameterized by $\boldsymbol{\theta} \in \mathbb{R}^p$: Let i.i.d. dataset of size m:

$$\mathfrak{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1,\ldots,m}.$$

Our loss function is CE, let loss on $\mathbf{x}_i$ and $\mathbf{y}_i$ be the:

$$\ell_i(\boldsymbol{\theta}) := \ell(f_{\boldsymbol{\theta}}(\mathbf{x}_i), \mathbf{y}_i).$$

Empirical loss function for the first k elements:

$$\mathcal{L}_k(\boldsymbol{\theta}) := \frac{1}{k}\sum_{i=1}^{k} \ell_i(\boldsymbol{\theta}), \quad \mathcal{L}(\boldsymbol{\theta}) := \mathcal{L}_m(\boldsymbol{\theta}).$$

Our objective is empirical loss function on the whole sample

$$\mathcal{L}_m = \frac{1}{m}\sum_{i=1}^{m} \ell_i(\boldsymbol{\theta}) \approx \mathbb{E}_{p(\mathbf{x},\mathbf{y})} \ell(f_{\boldsymbol{\theta}}(\mathbf{x}_i), \mathbf{y}_i).$$

One of our goals will be to estimate this difference.

$$\mathcal{L}_{k+1}(\boldsymbol{\theta}) - \mathcal{L}_k(\boldsymbol{\theta}) = \frac{1}{k+1}\big(\ell_{k+1}(\boldsymbol{\theta}) - \mathcal{L}_k(\boldsymbol{\theta})\big).$$

We introduce several definitions related to derivatives. Jacobian of the NN:

$$J(\boldsymbol{\theta}) := J_{f_{\boldsymbol{\theta}}(\mathbf{x})} = (\nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\mathbf{x}))^\intercal.$$

The total Hessian is written as:

$$\mathbf{H}^{(k)}(\boldsymbol{\theta}) := \nabla^2_{\boldsymbol{\theta}} \mathcal{L}_k(\boldsymbol{\theta}) = \frac{1}{k}\sum_{i=1}^{k} \nabla^2_{\boldsymbol{\theta}} \ell_i(\boldsymbol{\theta}).$$

B. Assumptions

To compare losses and arrange them in one point it would be convenient to introduce the following assumption.

Assumption 1. *Let $\boldsymbol{\theta}^*$ be the local minimum of both $\mathcal{L}_k(\boldsymbol{\theta})$ and $\mathcal{L}_{k+1}(\boldsymbol{\theta})$. In particular, it means, that $\nabla_{\boldsymbol{\theta}} \mathcal{L}_k(\boldsymbol{\theta}^*) = \nabla_{\boldsymbol{\theta}} \mathcal{L}_{k+1}(\boldsymbol{\theta}^*) = 0$.*

This assumption allows us to study the behavior of the landscape using only one point.

C. Approximation and Decomposition

Using this assumption and second-order approximation, in the work [34] it is shown, that to study local behavior one can use second-order Taylor approximation, from which we get

$$\begin{aligned}\big|\mathcal{L}_{k+1}(\boldsymbol{\theta}) - \mathcal{L}_k(\boldsymbol{\theta})\big| &\approx \frac{1}{k+1}\big|\ell_{k+1}(\boldsymbol{\theta}^*) - \mathcal{L}_k(\boldsymbol{\theta}^*)\big| + \\ &+ \frac{1}{k+1}\left\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\right\|^2 \left\|\mathbf{H}_{k+1}(\boldsymbol{\theta}^*) - \frac{1}{k}\sum_{i=1}^{k}\mathbf{H}_i(\boldsymbol{\theta}^*)\right\|. \quad (1)\end{aligned}$$

It is well known [36], that via chain rule, we can decompose out Hessian: Let $\mathbf{H}$ be the Hessian of ℓ on object $\mathbf{x}$, then

$$\begin{aligned}\mathbf{H} = \mathbf{H}_O + \mathbf{H}_F = J(\boldsymbol{\theta})^\intercal \left[\nabla^2_{f_{\boldsymbol{\theta}}} \ell(\boldsymbol{\theta})\right] J(\boldsymbol{\theta}) + \\ + \sum_{c=1}^{K} \left[\nabla_{f_{\boldsymbol{\theta}}} \ell(\boldsymbol{\theta})\right]_c \nabla^2_{\boldsymbol{\theta}} f^c_{\boldsymbol{\theta}}(\mathbf{x}).\end{aligned}$$

As [24], [37] can use only $\mathbf{H}_O$, because, at a point close to a local minimum, the average gradient is close to zero, then we can neglect the $\mathbf{H}_F$ term. Based on this approximation we will consider the norm of $\mathbf{H}_O$ term matrix. So, in terms of the matrix norm

$$\|\mathbf{H}\| \approx \left\|J(\boldsymbol{\theta})^\intercal \left[\nabla^2_{f_{\boldsymbol{\theta}}} \ell(\boldsymbol{\theta})\right] J(\boldsymbol{\theta})\right\|. \quad (2)$$

D. Outer-product Hessian

We adopt the term «outer-product» Hessian for the $\mathbf{H}_O$ term, as in [38]. As we will see later, it is in this form that it is most convenient to analyze the Hessian. In particular, we note that $\nabla^2_{f_{\boldsymbol{\theta}}(\boldsymbol{\theta})} \ell(\boldsymbol{\theta})$ depends only on loss function, for example:
for MSE loss: $\nabla^2_{f_{\boldsymbol{\theta}}(\boldsymbol{\theta})} \ell(\boldsymbol{\theta}) = I$
for CE(cross-entropy) loss: $\nabla^2_{f_{\boldsymbol{\theta}}(\boldsymbol{\theta})} \ell(\boldsymbol{\theta}) = diag(\mathbf{p}) - \mathbf{p}\mathbf{p}^\intercal$, where $p := \mathrm{SoftMax}(\mathbf{z})$. The choice of loss function is inconsequential, affecting only a multiplicative constant that

does not influence the overall analysis. We estimate the norm of matrix products by considering the product of their individual norms, leading to a quadratic dependence of the Hessian norm on the Jacobian in approximate calculations. As noted in [39], the Jacobian provides valuable structural insights into the network, which we examine further.

IV. MATRIX-PRODUCT NETWORK REPRESENTATION

Let $f_{\boldsymbol{\theta}}(\mathbf{x})$ be the composition of $L+1$ layers with ReLU activations

$$f_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{T}^{(L+1)} \circ \sigma \circ \dots \circ \sigma \circ \mathbf{T}^{(1)}(\mathbf{x}).$$

$\mathbf{T}^{(p+1)}$ — linear operator (or its matrix), σ — ReLU activation, the intermediate results can be represented as

$$\begin{cases} \mathbf{z}^{(p+1)} = \mathbf{T}^{(p+1)}\mathbf{x}^{(p)}, \\ \mathbf{x}^{(p+1)} = \sigma(\mathbf{z}^{(p+1)}) \end{cases}$$

with output logits $f_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{z} := \mathbf{z}^{(L+1)}$, and input $\mathbf{x}^{(0)} := \mathbf{x}$. Let $\mathbf{\Lambda}^{(p+1)} := diag(\mathbf{x}^{(p+1)} > 0)$ be input-dependent matrix. Then $f_{\boldsymbol{\theta}}(x)$ has the form of a product of matrices (m.b. input-dependent)

$$f_{\boldsymbol{\theta}}(x) = \mathbf{T}^{(L+1)}\mathbf{\Lambda}^{(L)} \dots \mathbf{\Lambda}^{(1)}\mathbf{T}^{(1)}\mathbf{x}. \tag{3}$$

The vector of weights is also considered $\boldsymbol{\theta} = col(\mathbf{W}^{(L+1)}, \dots, \mathbf{W}^{(1)})$, where $\mathbf{T}^{(p)}$ id differentiable and parameterized by part $\mathbf{W}^{(p)}$. Then the derivative of a layer with respect to its parameters can be determined.

$$\mathbf{Q}^{(p)} := \frac{\partial \mathbf{T}^{(p)}}{\partial \mathbf{W}^{(p)}},$$
$$\mathbf{Q} := diag(\mathbf{Q}^{(1)}, \dots, \mathbf{Q}^{(L+1)})$$

Where $\mathbf{Q}^{(p)}$ matrix gives a complete description of how the parameters are arranged in the p-th layer.
To simplify further formulas we define

$$\mathbf{G}^{(p)} := \mathbf{T}^{(L+1)}\mathbf{\Lambda}^{(L)} \dots \mathbf{T}^{(p+1)}\mathbf{\Lambda}^{(p)}; \mathbf{G}^{(L+1)} := \mathbf{I}$$
$$\mathbf{R}^{(p)} := \mathbf{\Lambda}^{(p)}\mathbf{T}^{(p)} \dots \mathbf{\Lambda}^{(1)}\mathbf{T}^{(1)}; \ p = \overline{1, L}; \ \mathbf{R}^{(0)} := \mathbf{I}.$$

Using notation, we can rewrite

$$\mathbf{z} = \mathbf{G}^{(p)}\mathbf{z}^{(p)}, \ \mathbf{x}^{(p)} = \mathbf{R}^{(p)}\mathbf{x},$$
$$\mathbf{z} = f_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{G}^{(p)}\mathbf{T}^{(p)}\mathbf{R}^{(p-1)}\mathbf{x}.$$

The stacked matrices $\mathbf{G}^{(p)}$ and $\mathbf{R}^{(p)}$ give us

$$\mathbf{F}^{\intercal} := \begin{pmatrix} \mathbf{G}^{(1)^{\intercal}} \otimes \mathbf{R}^{(0)}\mathbf{x} \\ \vdots \\ \mathbf{G}^{(k)^{\intercal}} \otimes \mathbf{R}^{(k-1)}\mathbf{x} \\ \vdots \\ \mathbf{G}^{(L+1)^{\intercal}} \otimes \mathbf{R}^{(L)}\mathbf{x} \end{pmatrix}.$$

The Hessian of a neural network by logits in case of CE loss function

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

where $\mathbf{p} := \mathrm{softmax}(\mathbf{z})$.

A. Hessian Structure

According to papers [19], [28], [34] we can decompose outer-product Hessian into simpler components, more specifically, we need to decompose only the Jacobian.

We will consider the key Lemmas of this paper, which describe the decomposition of the Hessian into a product of 5 matrices and the use of this representation to estimate the norm, the proofs are given in the appendix A and B.

Lemma 1. *Let our net $f_{\boldsymbol{\theta}}(\mathbf{x})$ can be represented as* (3)*, then* $\mathbf{H}_O(\boldsymbol{\theta}) = \mathbf{Q}^{\intercal}\mathbf{F}^{\intercal}\mathbf{A}\mathbf{F}\mathbf{Q}$.

Lemma 2. *Let the net $f_{\boldsymbol{\theta}}(\mathbf{x})$ can be represented as* (3)
Let $\forall p: \ \left\|\mathbf{Q}^{(p)}\right\| \leqslant q, \ \left\|\mathbf{T}^{(p)}\right\|^2 \leqslant w^2_{\mathbf{T}}$.
Then we have:
$\|\mathbf{H}_O\| \leqslant \sqrt{2}q^2 \left\|\mathbf{x}\right\|^2 (L+1)w^{2L}_{\mathbf{T}}$.

These lemmas are used to estimate the Hessian norms in special cases.

V. CONVOLUTIONS

A. 1D convolution

In this section, for simplicity, we keep the notation $\mathbf{T}^{(p)}$, but use it for 1D-convolutions, and we will explain how they are represented as linear operators. It is well known that convolutional networks can often be represented by a linear convolutional neural network (LCN). This usually refers to the Toeplitz representation of CNNs. [40], [41]. In this paper we use the notation for Toeplitz matrices from [19]. Also in the paper the authors found a specific type of matrix $\mathbf{Q}^{(p)}$, according to the structure of 1D Toeplitz matrix. Our 1D convolutional network is $f_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{T}^{(L+1)} * (\sigma(\dots(\sigma(\mathbf{T}^{(1)} * \mathbf{x}))\dots)$, where operation $*$ means convolution.
Let C_p be the number of channels after the p-th layer, and d_p be the size of sequence. There $\mathbf{x}^{(p)} \in \mathbb{R}^{C_p \times d_p}$, $\mathbf{T}^{(p)}$-1D convolution layer with kernel $\mathbf{W}^{(p)} \in \mathbb{R}^{C_{p-1} \times C_p \times k_p}$. To simplify further notations we can replace $\mathbf{x}^{(p)}$ with $vec(\mathbf{x}^{(p)}) \in \mathbb{R}^{(C_p d_p)}$. Now we have:

$$\mathbf{z}^{(p+1)} = \mathbf{T}^{(p+1)}\mathbf{x}^{(p)}.$$

The main results for 1D convolutions, which uses Toeplitz matrices to calculate $\mathbf{T}^{(p)}$ and $\mathbf{Q}^{(p)}$ like in [19] (out notation for the convolutions and Toeplitz matrices are identified for simplicity), and Lemmas 1 and 2. For more details see the proof in the appendix: C.

Theorem 1. *Consider the net $f_{\boldsymbol{\theta}}(x) = C_{\mathbf{W}^{(L+1)}} \circ \sigma \circ \dots \circ \sigma \circ C_{\mathbf{W}^{(1)}}$, where $C_{\mathbf{W}^{(i)}}$ - 1D convolution with kernel $\mathbf{W}^{(i)}$, without padding and with stride=1. Let the following upper bounds be given: $C_l \leqslant C$, $k_i \leqslant k$, $d_i \leqslant d_1 := d$, $|\mathbf{W}^{(p)}_{i,j,k}|^2 \leqslant w^2$. Then we can estimate outer-product hessian norm*

$$\|\mathbf{H}_O\| \leqslant \sqrt{2}\left\|x\right\|^2 d^2(L+1)(C^2w^2kd)^L.$$

As investigation, we apply this theorem to the difference of loss, same as [34].

Corollary 1. *let $\boldsymbol{\theta}$ be in the R-vicinity of optima: $\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\| \leqslant R$. Loss function is limited by some constant: $\exists\ W_l > 0 :\ \forall i\ |\ell_i| \leqslant W_l$. Let all objects in the dataset are limited too: $\exists W_x\ \forall i\ \|x_i\| \leqslant W_x$. Then, under the conditions of the theorem 1 and our assumptions, we have:*

$$\left|\mathcal{L}_{k+1}(\boldsymbol{\theta}) - \mathcal{L}_k(\boldsymbol{\theta})\right| \leqslant \frac{2}{k+1}W_\ell + \\ + \frac{2}{k+1}R^2\sqrt{2}d^2W_x^2(L+1)(C^2w^2kd)^L.$$

As we can see, this estimate is extremely high compared to the actual norm. However, one can hypothesize that the dependence on the number of channels, the weight norms, and the sizes is indeed the same as presented above. We suppose that if the parameters for convolution are already sufficient for training, then increasing, for example, the size of the convolution kernel k will unjustifiably increase the estimate ,approximately by a factor of $(1 + \frac{\Delta k}{k})^L$, which can lead to a slowdown in the rate of convergence without significant improvements in quality.

B. 2D-convolution

In this section we consider 2D convolutional networks. Again, for simplicity, we retain the $\mathbf{T}^{(p)}$ notation for convolutional network layers, and we will explain why convolution can be considered a linear layer. $\mathbf{x} \in \mathbb{R}^{m \times n \times C}$ - input image, which has (m, n) dimentions and C channels.
$\mathbf{x}^{(l)} \in \mathbb{R}^{m_i \times n_i \times C_i}$ - input of $l+1$-th layer. $\mathbf{W}^{(l)} \in \mathbb{R}^{C_{l-1} \times C_l \times k_l^1 \times k_l^2}$ - convolution with kernel sizes (k_l^1, k_l^2), input and output numbers of channels C_{l-1}, C_l respectively. Similar to the section A, we use $vec(\mathbf{x}) \in \mathbb{R}^{m_i n_i C_i}$ instead of $\mathbf{x} \in \mathbb{R}^{m_i \times n_i \times C_i}$. The operation of convolution on the input tensor is examined, particularly in the case of a vectorized input. We can use the same Toeplitz framework as in [42], but it's easier for us to use a specific matrix $\mathbf{T}^{(p)}$, a row of which consists of elements $\mathbf{W}^{(p)}_{*,c_2,*,*}$ for the c_2-th channel. That is, each row of $\mathbf{T}^{(p)}$ implements "applying" the kernel to a specific entry position and to a specific channel. Let $\mathbf{T}_i^{(p)}$ match the $c_2 = c_2(i)$-th channel of $\mathbf{W}$.

Now we can formulate a theorem about the Hessian norm of a convolution; you will find the proof in the appendix D.

Theorem 2. *Let the net $f_{\boldsymbol{\theta}}(\mathbf{x}) = C_{\mathbf{W}^{(L+1)}} \circ \cdots \circ C_{\mathbf{W}^{(1)}}$, where $C_{\mathbf{W}^{(l)}}$ - 2D convolution with kernel $\mathbf{W}^{(i)}$, without padding and with stride=1. Also let the following upper bounds be given: $C_l \leqslant C$, $k_i \leqslant k$, $m_i \leqslant m_1 := m$, $n_i \leqslant n_1 := n$, $|\mathbf{W}^{(p)}_{i,j,k}|^2 \leqslant w^2$, Then the hessian norm*

$$\|\mathbf{H}_O\| \leqslant \sqrt{2}\,\|\mathbf{x}\|^2\, q^2(L+1)(C^2k^2w^2mn)^L,$$

where $q^2 = C^2k^2mn$.

We can evaluate the difference in loses, under out assmptions and approximations.

Corollary 2. *Let $\boldsymbol{\theta}$ be in the R-vicinity of optima: $\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\| \leqslant R$
Also loss function is limited by some constant: $\exists\ W_l > 0 :\ \forall i\ |\ell_i| \leqslant W_l$. Let all objects in dataset is limited too: $\exists W_x\ \ \forall i\ \ \|x_i\| \leqslant W_x$. Then, under the conditions of the theorem 2, and our assumptions, we have:*

$$\left|\mathcal{L}_{k+1}(\boldsymbol{\theta}) - \mathcal{L}_k(\boldsymbol{\theta})\right| \leqslant \frac{2}{k+1}W_\ell + \\ + \frac{2}{k+1}R^2\sqrt{2}q^2W_x^2(L+1)(C^2k^2w^2mn)^L,$$

where $q^2 = C^2k^2mn$.

As can be seen, the network in this example consists solely of convolutional layers, which is a rare occurrence in practice. In 5, we discussed the case of adding a fully connected head to a convolutional neural network. These results allow us to construct a hypothesis that the Hessian norm can be an exponential function of the number of layers and also depends on kernel size, image dimensions, and channels in the manner described above. The major disadvantage of these results is that they are not affected by the reduction in sizes after convolutions and depend only on the upper bounds of the parameters.

C. Poolings

We also provide results related to adding pooling to the network First is about max pooling, proof one can find in the appendix E.

Lemma 3. *Let our convolutional net $f_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{T}^{(L+1)}\boldsymbol{\Lambda}^{(L)} \dots \boldsymbol{\Lambda}^{(1)}\mathbf{T}^{(1)}\mathbf{x}$
contains a MaxPool2D in layer $\boldsymbol{\Lambda}^{(l)}$ with kernel $k_{\text{pool}} \times k_{\text{pool}}$ instead of ReLU activation . Then $\|\mathbf{H}_O\| \leqslant \sqrt{2}\,\|\mathbf{x}\|^2\, q^2\big(\frac{1}{k^2_{\text{pool}}}\big)^{L-l+2}(L+1)(k^2C^2w^2mn)^L$, where $q^2 = mnC^2k^2$.*

And the result is about adding average pooling, proof one can find there F

Lemma 4. *Let our conv net
$f_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{T}^{(L+1)}\boldsymbol{\Lambda}^{(L)} \dots \boldsymbol{\Lambda}^{(1)}\mathbf{T}^{(1)}\mathbf{x}$
contains AvgPool2D in layer $\boldsymbol{\Lambda}^{(l)}$ instead of ReLU activation with kernel of size $k_{\text{pool}} \times k_{\text{pool}}$. Then*

$$\|\mathbf{H}_O\| \leqslant \sqrt{2}\,\|\mathbf{x}\|^2\, q^2\big(\frac{1}{k^2_{\text{pool}}}\big)^{L-l+2}(L+1)(k^2C^2w^2mn)^L,$$

where $q^2 = mnC^2k^2$.

D. Fully connected head

One can see, that our network consisted exclusively of convolutional layers, which almost never happens, consider a network that appears fully connected in its last P layers. Proof is in the appendix G.

Lemma 5. *Let our conv network with classification head of size P is*

$$f_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{T}^{(L+P+1)}\boldsymbol{\Lambda}^{(L+P)} \dots \\ \dots \boldsymbol{\Lambda}^{(L+1)}\mathbf{T}^{(L+1)}\boldsymbol{\Lambda}^{(L)} \dots \boldsymbol{\Lambda}^{(1)}\mathbf{T}^{(1)}\mathbf{x},$$

where $\mathbf{T}^{(L+1+i)}$ *- Linear layers with* h_i *parameters where* $i = 1, \ldots, P$, $\mathbf{T}^{(r)}$*-2D-conv layers as in V-B. We suppose that* $\left\|\mathbf{T}_{ij}^{(L+1+i)}\right\| \leqslant \tilde{w}$ *and* $h_p \leqslant h$*. Then, under the conditions and notations of Theorem 2, we have*

$$\|\mathbf{H}_O\| \leqslant \sqrt{2}\,\|\mathbf{x}\|^2\, q^2 (h^2 \tilde{w}^2)^P (k^2 C^2 w^2 mn)^L \times$$
$$\times \left(L + 1 + P \frac{h^2 \tilde{w}^2}{k^2 C^2 w^2 mn}\right).$$

VI. Experiments

To validate the theoretical estimates, we conducted a comprehensive empirical investigation. In this section, we present the findings from training convolutional networks with different parameters.

The primary purpose of the experiments is to demonstrate the dependence of the loss function landscape on parameters such as the number of layers, kernel size, number of channels, pooling positions, and to observe how the convergence rate depends on these parameters. To achieve this, we trained convolutional networks and obtained parameters $\hat{\boldsymbol{\theta}}$ near the optimum. We used a convolutional architecture with ReLU activation after each layer. To trace the influence of a specific parameter on convergence, we fixed the key parameters of the neural network, varied the hyperparameter of interest and trained a corresponding set of models.

Then, we examined the relationship between the average absolute difference between the average loss function values and the available sample size. Next, for each model, to obtain more robust results, we averaged the loss difference across shuffled samples. Additionally, for enhanced visualization, we employed exponential smoothing with a coefficient of 0.995.

For this study, we employed the numerical representation of image pixels as our input data. The results derived from examining samples within the MNIST [43], FashionMNIST [44] and CIFAR10 [45] database.

In all experiments, the following hyperparameters were: constant learning rate of 1e-3, Adam optimizer, we used mini-batches of size 64, we trained for 10 epochs on the MNIST and Fashion-MNIST datasets, and 15 epochs on the CIFAR-10 dataset. If a parameter was not varied, it was kept the same across all layers.

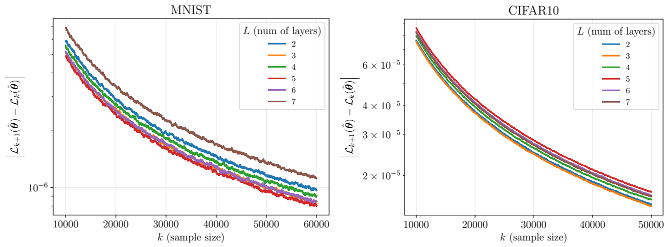

Fig. 1: The variable number of hidden convolutional layers L with fixed kernel size $k = 3$ and number of channels $C = 6$. Analysis of the resultant graphs reveals a non-monotonic relationship between the output values and the number of layers.

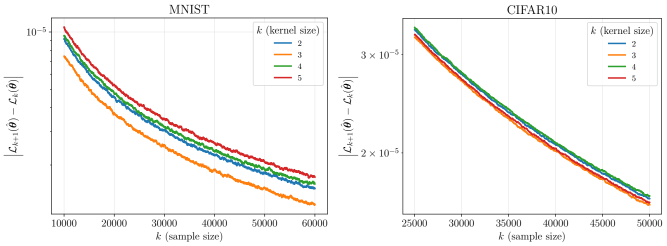

Fig. 2: The variable the kernel size k with fixed number of convolutional layers L and number of channels $C = 6$. The data exhibit a non-monotonic relationship with respect to kernel size.

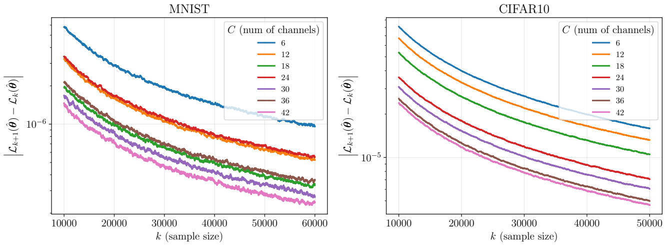

Fig. 3: The variable channel num C with fixed number of convolutional layers L and kernel size $k = 3$. The dependence of the value on the number of channels is monotonic.

VII. Discussion

As demonstrated by the graphs, the absolute difference between the average loss function values does not directly depend on kernel size or the number of network layers. However, it shows a monotonic relationship with layer size and pooling position. We suppose that this primarily indicates that the first part of Equation (1) has a more significant influence on this value. Consequently, it suggests that the loss function value at the optimum point is the predominant factor affecting this relationship.

It is noteworthy that the experiments employed relatively small networks, which means that increasing the number of parameters enhanced model quality, substantially influencing the results. In particular on the value of the loss function at optimum.

We have identified several potential solutions to this problem. First, we propose examining more complex network structures where increasing the parameter count would not have such a significant impact on our estimates.

Furthermore, it is evident that our estimation significantly exceeds realistic values and primarily serves as a theoretical construct rather than a practical measure. Primarily, the substantial overestimation can be attributed to the fact that the norm of matrix $\mathbf{T}^{(p)}$ was evaluated through the product of norms (see proofs C or D), This approach, when applied to our specific case involving sparse matrices, inevitably leads to significantly overestimated results. The sparsity of the matrices in question exacerbates the discrepancy between the estimated and actual norms.

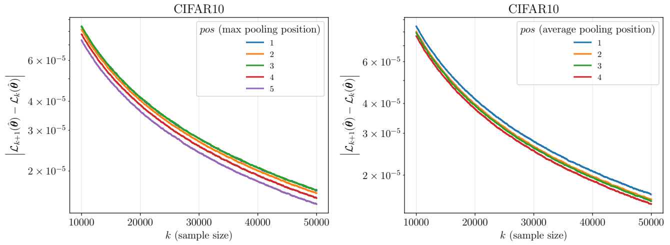

Fig. 4: The variable channels num C with fixed number of convolutional layers L and kernel size $k = 3$. The graph shows a monotonic dependence of the value depending on the position of the pooling in the network.

We posit that our research has potential applications in several areas, including exploring the functional landscape through analysis of the loss function's Hessian, developing techniques for determining appropriate sample sizes, and investigating the structural properties of neural network Hessian. These applications could contribute to a deeper understanding of neural network behavior.

VIII. Conclusion

In this paper we proposed a method for estimating the Hessian norm, and also we have suggested a way to utilize this norm to estimate the convergence of the loss landscape. Using a second-order approximation of the loss function, our theoretical analysis suggests demonstrated how the convergence of the loss function landscape can depend on the norm of the Hessian, and also how the norm of the Hessian can depend on the parameters of the network. Our empirical results showed that the dependence of the absolute difference between the average loss function values has an ambiguous nature. We believe, that our results provide valuable insights into the local geometry of loss landscapes and properties of network hessian.

References

[1] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. de Las Casas, L. A. Hendricks, J. Welbl, A. Clark, T. Hennigan, E. Noland, K. Millican, G. van den Driessche, B. Damoc, A. Guy, S. Osindero, K. Simonyan, E. Elsen, J. W. Rae, O. Vinyals, and L. Sifre, "Training compute-optimal large language models," 2022. [Online]. Available: https://arxiv.org/abs/2203.15556

[2] A. Grabovoy, O. Bakhteev, and V. Strijov, "Estimation of the relevance of the neural network parameters," *Informatics and Applications*, Jun. 2019. [Online]. Available: http://dx.doi.org/10.14357/19922264190209

[3] ——, "Ordering the set of neural network parameters," *Informatics and Applications*, Jun. 2020. [Online]. Available: http://dx.doi.org/10.14357/19922264200208

[4] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, "Visualizing the loss landscape of neural nets," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2018/file/a41b3bb3e6b050b6c9067c67f663b915-Paper.pdf

[5] L. Wang, B. Shen, N. Zhao, and Z. Zhang, "Is the skip connection provable to reform the neural network loss landscape?" in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, ser. IJCAI'20, 2021.

[6] A. S. Bosman, A. Engelbrecht, and M. Helbig, "Empirical loss landscape analysis of neural network activation functions," in *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*, ser. GECCO '23 Companion, vol. 33. ACM, 2023. [Online]. Available: http://dx.doi.org/10.1145/3583133.3596321

[7] M. H. Anna Sergeevna Bosman, Andries Engelbrecht, "Visualising basins of attraction for the cross-entropy and the squared error neural network loss functions," *Neurocomputing*, vol. 400, pp. 113–136, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231220303593

[8] B. Neyshabur, Z. Li, S. Bhojanapalli, Y. LeCun, and N. Srebro, "The role of over-parametrization in generalization of neural networks," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=BygfghAcYX

[9] D. Zou and Q. Gu, "An improved analysis of training over-parameterized deep neural networks," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/6a61d423d02a1c56250dc23ae7ff12f3-Paper.pdf

[10] Y. Lee, J. R. Willette, J. Kim, and S. J. Hwang, "Visualizing the loss landscape of self-supervised vision transformer," 2024. [Online]. Available: https://arxiv.org/abs/2405.18042

[11] X. Chen, C.-J. Hsieh, and B. Gong, "When vision transformers outperform resnets without pre-training or strong data augmentations," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=LtKcMgGOeLt

[12] M. Elhamod and A. Karpatne, "Neuro-visualizer: An auto-encoder-based loss landscape visualization method," 2023. [Online]. Available: https://arxiv.org/abs/2309.14601

[13] R. Bain, "Visualizing the loss landscape of winning lottery tickets," 2021. [Online]. Available: https://arxiv.org/abs/2112.08538

[14] Q. Yuan and N. Xiao, "Experimental exploration on loss surface of deep neural network," *International Journal of Imaging Systems and Technology*, vol. 30, no. 4, pp. 860–873, 2020. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/ima.22434

[15] D. J. Im, M. Tao, and K. Branson, "An empirical analysis of the optimization of deep network loss surfaces," 2017. [Online]. Available: https://arxiv.org/abs/1612.04010

[16] B. Ghorbani, S. Krishnan, and Y. Xiao, "An investigation into neural net optimization via hessian eigenvalue density," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 2232–2241. [Online]. Available: https://proceedings.mlr.press/v97/ghorbani19b.html

[17] V. Papyan, "The full spectrum of deep net hessians at scale: Dynamics with sample size," *CoRR*, vol. abs/1811.07062, 2018. [Online]. Available: http://arxiv.org/abs/1811.07062

[18] Z. Liao and M. W. Mahoney, "Hessian eigenspectra of more realistic nonlinear models," in *Proceedings of the 35th International Conference on Neural Information Processing Systems*, ser. NIPS '21. Red Hook, NY, USA: Curran Associates Inc., 2024.

[19] S. P. Singh, T. Hofmann, and B. Schölkopf, "The hessian perspective into the nature of convolutional neural networks," in *Proceedings of the 40th International Conference on Machine Learning*, ser. ICML'23. JMLR.org, 2023.

[20] A. Grabovoy, T. Gadaev, A. Motrenko, and V. Strijov, "Numerical methods of sufficient sample size estimation for generalised linear models," *Lobachevskii Journal of Mathematics*, vol. 43, pp. 2453–2462, 12 2022.

[21] A. Azadbakht, S. R. Kheradpisheh, I. Khalfaoui-Hassani, and T. Masquelier, "Drastically reducing the number of trainable parameters in deep cnns by inter-layer kernel-sharing," 2022. [Online]. Available: https://arxiv.org/abs/2210.14151

[22] A. A. Kroshchanka, V. A. Golovko, and M. Chodyka, "Method for reducing neural-network models of computer vision," *Pattern Recognit. Image Anal.*, vol. 32, no. 2, p. 294–300, Jun. 2022. [Online]. Available: https://doi.org/10.1134/S1054661822020146

[23] K. Kahatapitiya and R. Rodrigo, "Exploiting the redundancy in convolutional filters for parameter reduction," in *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2021, pp. 1409–1419.

[24] S. Fort and S. Ganguli, "Emergent properties of the local geometry of neural loss landscapes," *CoRR*, vol. abs/1910.05929, 2019. [Online]. Available: http://arxiv.org/abs/1910.05929

[25] S. P. Singh, L. Adilova, M. Kamp, A. Fischer, B. Schölkopf, and T. Hofmann, "Landscaping linear mode connectivity," 2024. [Online]. Available: https://arxiv.org/abs/2406.16300

[26] X.-C. Li, L. Li, and D.-C. Zhan, "Visualizing, rethinking, and mining the loss landscape of deep neural networks," 2024. [Online]. Available: https://arxiv.org/abs/2405.12493

[27] S. P. Singh, A. Lucchi, T. Hofmann, and B. Schölkopf, "Phenomenology of double descent in finite-width neural networks," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=lTqGXfn9Tv

[28] Y. Wu, X. Zhu, C. Wu, A. N. Wang, and R. Ge, "Dissecting hessian: Understanding common structure of hessian in neural networks," 2021. [Online]. Available: https://openreview.net/forum?id=0rNLjXgchOC

[29] M. Skorski, "Chain rules for hessian and higher derivatives made easy by tensor calculus," 2019. [Online]. Available: https://arxiv.org/abs/1911.13292

[30] V. Papyan, "The full spectrum of deepnet hessians at scale: Dynamics with sgd training and sample size," 2019. [Online]. Available: https://arxiv.org/abs/1811.07062

[31] L. Sagun, L. Bottou, and Y. LeCun, "Eigenvalues of the hessian in deep learning: Singularity and beyond," 2017. [Online]. Available: https://arxiv.org/abs/1611.07476

[32] V. Papyan, "Traces of class/cross-class structure pervade deep learning spectra," *Journal of Machine Learning Research*, vol. 21, no. 252, pp. 1–64, 2020. [Online]. Available: http://jmlr.org/papers/v21/20-933.html

[33] ——, "Measurements of three-level hierarchical structure in the outliers in the spectrum of deepnet hessians," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 5012–5021. [Online]. Available: https://proceedings.mlr.press/v97/papyan19a.html

[34] N. Kiselev and A. Grabovoy, "Unraveling the hessian: A key to smooth convergence in loss function landscapes," 2024. [Online]. Available: https://arxiv.org/abs/2409.11995

[35] J. R. Magnus and H. Neudecker, *Matrix Differential Calculus with Applications in Statistics and Econometrics*, 2nd ed. John Wiley, 1999.

[36] N. N. Schraudolph, "Fast curvature matrix-vector products for second-order gradient descent," *Neural Comput.*, vol. 14, no. 7, p. 1723–1738, Jul. 2002. [Online]. Available: https://doi.org/10.1162/08997660260028683

[37] L. Sagun, U. Evci, V. U. Güney, Y. N. Dauphin, and L. Bottou, "Empirical analysis of the hessian of over-parametrized neural networks," *CoRR*, vol. abs/1706.04454, 2017. [Online]. Available: http://arxiv.org/abs/1706.04454

[38] F. Latrémolière, S. Narayanappa, and P. Vojtěchovský, "Estimating the jacobian matrix of an unknown multivariate function from sample values by means of a neural network," 2022. [Online]. Available: https://arxiv.org/abs/2204.00523

[39] S. Hayou, B. Dadoun, P. Youssef, H. Salam, and M. E. A. Seddik, "A theoretical study of the jacobian matrix in deep neural networks," 2024. [Online]. Available: https://openreview.net/forum?id=pvhyBB86Bt

[40] K. Kohn, T. Merkh, G. Montúfar, and M. Trager, "Geometry of linear convolutional networks," *SIAM Journal on Applied Algebra and Geometry*, vol. 6, no. 3, pp. 368–406, 2022. [Online]. Available: https://doi.org/10.1137/21M1441183

[41] Z. Qin, X. Han, W. Sun, B. He, D. Li, D. Li, Y. Dai, L. Kong, and Y. Zhong, "Toeplitz neural network for sequence modeling," 2023. [Online]. Available: https://arxiv.org/abs/2305.04749

[42] M. Gnacik and K. Łapa, "Using toeplitz matrices to obtain 2d convolution," 10 2022.

[43] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Processing Magazine*, vol. 29, pp. 141–142, 2012. [Online]. Available: https://api.semanticscholar.org/CorpusID:5280072

[44] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," 2017, cite arxiv:1708.07747Comment: Dataset is freely available at https://github.com/zalandoresearch/fashion-mnist Benchmark is available at http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/. [Online]. Available: http://arxiv.org/abs/1708.07747

[45] A. Krizhevsky, "Learning multiple layers of features from tiny images," 2009. [Online]. Available: https://api.semanticscholar.org/CorpusID:18268744

Appendix A
Proof of Lemma 1

Proof. Output of our convolution is logits:

$$\mathbf{z} = f_{\boldsymbol{\theta}}(x) = \mathbf{T}^{(L+1)}\boldsymbol{\Lambda}^{(L)}\mathbf{T}^{(L)}....\boldsymbol{\Lambda}^{(1)}\mathbf{T}^{(1)}\mathbf{x}.$$

We will work with derivative with respect to parameters

$\dfrac{\partial \mathbf{z}}{\partial \mathbf{W}^{(p)}} = \dfrac{\partial \mathbf{z}}{\partial \mathbf{z}^{(p)}}\dfrac{\partial \mathbf{z}^{(p)}}{\partial \mathbf{T}^{(p)}}\dfrac{\partial \mathbf{T}^{(p)}}{\partial \mathbf{W}^{(p)}}$ as a Jacobian of composition

Use $vec(\mathbf{B}\mathbf{V}\mathbf{A}^\intercal) = (\mathbf{A} \otimes \mathbf{B})vec(\mathbf{V})$ with $\mathbf{A} = \mathbf{I}$ and get vectorized $\mathbf{z}^{(p)}$

$$vec(\mathbf{z}^{(p)}) = vec(\mathbf{T}^{(p)}\mathbf{x}^{(p-1)}) = (\mathbf{I} \otimes \mathbf{x}^{(p-1)})vec(\mathbf{T}^{(p)}).$$

After one can see, that

$$\frac{\partial \mathbf{z}^{(p)}}{\partial \mathbf{T}^{(p)}} = \mathbf{I} \otimes \mathbf{x}^{(p-1)^\intercal}.$$

From $\mathbf{z} = \mathbf{G}^{(p)}\mathbf{z}^{(p)}$ we achieve

$$\frac{\partial \mathbf{z}}{\partial \mathbf{z}^{(p)}} = \mathbf{G}^{(p)}.$$

The definition of $\mathbf{Q}^{(p)}$:

$$\frac{\partial \mathbf{T}^{(p)}}{\partial \mathbf{W}^{(p)}} = \mathbf{Q}^{(p)}.$$

Using

$$\mathbf{A}_i \in \mathbb{R}^{m_i \times n_i} \text{ then } \mathbf{A}_1 \otimes \mathbf{A}_2 = (\mathbf{A}_1 \otimes \mathbf{I}_{m_2})(\mathbf{I}_{m_1} \otimes \mathbf{A}_2).$$

with $m_2 = 1$ we get

$$\mathbf{G}^{(p)}\left(\mathbf{I}\otimes\mathbf{x}^{(p-1)^\intercal}\right) = \left(\mathbf{G}^{(p)}\otimes\mathbf{I}_1\right)\left(\mathbf{I}\otimes\mathbf{x}^{(p-1)^\intercal}\right) = \mathbf{G}^{(p)}\otimes\mathbf{x}^{(p)^\intercal}.$$

we substitute the above statements into one formula and get

$$\frac{\partial \mathbf{z}}{\partial \mathbf{W}^{(p)}} = (\mathbf{G}^{(p)}\otimes\mathbf{I}_1)(\mathbf{I}\otimes\mathbf{x}^{(p-1)^\intercal})\mathbf{Q}^{(p)} = (\mathbf{G}^{(p)}\otimes\mathbf{x}^{(p)^\intercal})\mathbf{Q}^{(p)}.$$

As in [19] consider the block $\mathbf{H}_O^{(kl)}$:

$$\begin{aligned} &\mathbf{H}_O^{(kl)} = J(\boldsymbol{\theta})^\intercal \mathbf{A} J(\boldsymbol{\theta}) = \\ &= \mathbf{Q}^{(k)^\intercal}(\mathbf{G}^{(k)^\intercal} \otimes \mathbf{R}^{(k-1)}\mathbf{x})A(\mathbf{G}^{(l)} \otimes \mathbf{x}^\intercal\mathbf{R}^{(l-1)^\intercal})\mathbf{Q}^{(l)} \end{aligned}$$

Then $\mathbf{H}_O = \mathbf{Q}^{\mathbf{T}}\mathbf{F}\mathbf{A}\mathbf{F}^{\mathbf{T}}\mathbf{Q}$.

□

Appendix B
Proof of Lemma 2

Proof. Using the results of the previous Lemma 1, it is enough for us to evaluate the upper bound of the expression: $\|\mathbf{Q}\|^2 \|\mathbf{F}\|^2 \|\mathbf{A}\|$
In the work [34], the norm of matrix $\mathbf{A}$ was examined, and it was proven that:

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

Norm of block-diagonal matrix is not greater than max of block's norm

$$\|\mathbf{Q}\|^2 \leqslant \max_{i=1,\dots,L+1} \left\|\mathbf{Q}^{(i)}\right\|^2 \leqslant q^2.$$

Norm of matrix product is less or equal then product of norms:

$$\left\|\mathbf{G}^{(p)}\right\|^2 \leqslant \left\|\mathbf{T}^{(p+1)}\right\|^2 \dots \left\|\mathbf{T}^{(L+1)}\right\|^2 \leqslant w_{\mathbf{T}}^{2(L-p+1)}.$$

$$\left\|\mathbf{R}^{(p-1)}\right\|^2 \leqslant \left\|\mathbf{T}^{(1)}\right\|^2 \dots \left\|\mathbf{T}^{(p-1)}\right\|^2 \leqslant w_{\mathbf{T}}^{2(p-1)}.$$

The spectral matrix norm of the Kronecker product is equal to their ordinary product norm. Spectral norm of vertical stacked matrices is less or equal then sum of norms of it's blocks

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

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

Substituting the obtained estimates into the $\|\mathbf{H}_O\|$ formula we get

$$\|F\|^2 \leqslant \|\mathbf{x}\|^2 \sum_{p=1}^{L+1} w_{\mathbf{T}}^{2L} \leqslant \|\mathbf{x}\|^2 (L+1) w_{\mathbf{T}}^{2L}.$$

$$\|\mathbf{H}_O\| \leqslant \|\mathbf{Q}\|^2 \|\mathbf{F}\|^2 \|\mathbf{A}\| \leqslant \sqrt{2}\, \|\mathbf{x}\|^2 q^2 (L+1) w_{\mathbf{T}}^{2L}.$$

□

Appendix C
Proof of Theorem 1

Proof. It is clear that, based on the Lemma 2, we need to proof only 2 statements:

$$\left\|\mathbf{T}^{(p)}\right\|^2 \leqslant C^2 dkw^2,$$

$$\left\|\mathbf{Q}^{(p)}\right\|^2 \leqslant d^2.$$

In [19], it is easy to see, that in $\mathbf{T}^{(p)}$ every block of $C_l C_{l-1}$ has d_{l-1} rows with kernel in the right position, which lead us to

$$\left\|\mathbf{T}^{(p)}\right\|^2 \leqslant C^2 dkw^2.$$

To prove the second inequality we again turn to [19] and estimate the norm of vertically stacked matrices:

$$\frac{\partial \mathbf{T}^{(l)}}{\partial \mathbf{W}^{(l)}} =: \mathbf{Q}^{(l)} = \mathbf{I}_{C_l} \otimes \begin{pmatrix} \mathbf{I}_{C_{l-1}} \otimes (\pi_R^0 \mathbf{I}_{d_{l-1} \times k_l}) \\ \vdots \\ \mathbf{I}_{C_{l-1}} \otimes (\pi_R^{d_{l-1}-k_l} \mathbf{I}_{d_{l-1} \times k_l}) \end{pmatrix}.$$

$$\left\|\mathbf{Q}^{(l)}\right\| \leqslant \sum_{i=0}^{d_{l-1}-k_l} \left\|\pi_R^i \mathbf{I}_{d_{l-1} \times k_l}\right\| \leqslant \sum_{i=1}^{d_{l-1}-k_l} \|\pi_R\| =$$

$$= \sum_{i=0}^{d_{l-1}-k_l} 1 = d_{l-1} - k_l + 1 = d_l \leqslant d_1 = d.$$

□

Appendix D
Proof of Theorem 2

Proof. from the description V-B of the matrix $\mathbf{T}^{(p)}$ one can see, that

$$\left\|\mathbf{T}_{i,*}^{(p)}\right\|^2 = \sum_{c,k,l}^{C_{p-1}, k_p^1, k_p^2} |\mathbf{W}_{c,c_2,k,l}^{(p)}|^2.$$

And as an obvious consequence

$$\left\|\mathbf{T}^{(p)}\right\|_F^2 = \sum_{c_1,i,k,l}^{C_{p-1}, C_p n_p m_p, k_p^1, k_p^2} \left(\mathbf{W}_{c_1,c_2(i),k,l}^{(p)}\right)^2. \tag{4}$$

Here we assume a simple correspondence between the output channel c_2 and the $\mathbf{T}^{(p)}$ line i.

By analogy with the proof C, using 2 we need to proof 2 statements:

$$\left\|\mathbf{T}^{(p)}\right\| \leqslant C^2 k^2 w^2 mn.$$

$$\left\|\mathbf{Q}^{(p)}\right\| \leqslant C^2 k^2 mn.$$

Initially, the norm of $\mathbf{T}^{(p)}$ is estimated:

$$\left\|\mathbf{T}^{(p)}\right\|^2 \leqslant \left\|\mathbf{T}^{(p)}\right\|_F^2 \leqslant |_{\text{use (4)}}| \leqslant \sum_i Ck^2 w^2 \leqslant C^2 k^2 w^2 mn.$$

Next, we will estimate norm of the layer derivative with respect to the parameters.

$$\left\|\mathbf{Q}^{(p)}\right\| = \left\|\frac{\partial \mathbf{T}^{(p)}}{\partial \mathbf{W}^{(p)}}\right\|.$$

As stated earlier the row of $\mathbf{T}^{(p)}$ - it is exactly $vec_r(\mathbf{W}_{*,i,*,*}^{(p)})$ arranged in the correct order. Then the norm of the row is: $\frac{\partial \mathbf{T}_{(i,j)}^{(p)}}{\partial \mathbf{W}_{c_1,c_2,k_1,k_2}^{(p)}} \neq 0 \iff$ the indices are selected in such a way, that $T_i^{(p)}$ corresponds c_2 and at the same time $\mathbf{T}_{i,j}^{(p)}$ corresponds c_1, k_1, k_2 and this correspondence depends on the specific matrix $\mathbf{T}^{(p)}$, but it is obvious, that one i corresponds to only one c_2, because every row participates in the formation of only one element of one channel. Since only $\mathbf{W}_{*,c_2,*,*}^{(p)}$ participates in the formation of one line $\mathbf{T}_{i,*}^{(p)}$, we can fixed i and corresponding c_2, and also at the same time we know that, for every c_1, k_1, k_2, there are only one column j: $\mathbf{T}_{i,j}^{(p)} = \mathbf{W}_{c_1,c_2,k_1,k_2}^{(p)}$:

$$\sum_{j,c_1,k_1,k_2} \left(\frac{\partial \mathbf{T}_{i,j}^{(p)}}{\partial \mathbf{W}_{c_1,c_2,k_1,k_2}^{(p)}}\right)^2 =$$

$$\sum_{c_1,k_1,k_2} \sum_j \left(\frac{\partial \mathbf{T}_{i,j}^{(p)}}{\partial \mathbf{W}_{c_1,c_2,k_1,k_2}^{(p)}}\right)^2 =$$

$$= |_{\text{in the inner sum there is only one non-zero term}}| =$$

$$= \sum_{c_1,k_1,k_2} 1 = C_{p-1} k_p^1 k_p^2 \leqslant Ck^2.$$

Consider the Frobenius norm as the upper bound of the spectral norm:

$$\|\mathbf{Q}\|^2 \leqslant \|\mathbf{Q}\|_F^2 = \sum_{i,j,c_1,c_2,k_1,k_2} \left(\frac{\partial \mathbf{T}_{i,j}^{(p)}}{\partial \mathbf{W}_{c_1,c_2,k_1,k_2}^{(p)}} \right)^2 =$$

$$= \sum_{i,c_2} \sum_{j,c_1,k_1,k_2} \left(\frac{\partial \mathbf{T}_{i,j}^{(p)}}{\partial \mathbf{W}_{c_1,c_2,k_1,k_2}^{(p)}} \right)^2 =$$

$$= \left|_{\text{estimated the internal sum earlier and only under appropriate } i \text{ and } c_2}\right| \leqslant$$

$$\leqslant \sum_i Ck^2 = CmnCk^2 \leqslant C^2k^2mn.$$

□

Appendix E
Proof of the Lemma 3

Proof. Let's use the notation $\mathbf{M}^{(l)}$ for our 2D-Max-Pool layer Like for convolutions, we can describe every row $\mathbf{M}^{(l)}$:
Before starting, there some properties of $\mathbf{M}$, that which will be used: At first, that the row $\mathbf{M}_{i*}$ corresponds to specific pooling window(to elements covered by window), and, as the second one, is that column $\mathbf{M}_{*j}$ corresponds to elements are multiplied by j'th element of input.
Since every window covers only one element and 2 different windows do not intersect, then there are only one element in every row, thus

$$\left\|\mathbf{M}^{(l)}\right\| = \sqrt{\lambda_{max}(\mathbf{M}^{(l)\intercal}\mathbf{M}^{(l)})} = 1,$$

because $(\mathbf{M}_{*,i}^{(l)}, \mathbf{M}_{*,j}^{(l)}) \neq 0 \iff (\mathbf{M}_{*,i}^{(l)}, \mathbf{M}_{*,j}^{(l)}) = 1 \iff i = j$ and i'th element is max in appropriate window. For simplicity, we assume that $\mathbf{M}^{(l)}$ reduces both dimentions k_{pool} times Similar to the D we estimate $\mathbf{G}^{(p)}$ and $\mathbf{R}^{(p-1)}$ component, however, in accordance with the new layer

$$\left\|\mathbf{G}^{(p)}\right\| \left\|\mathbf{R}^{(p-1)}\right\| \leqslant \frac{\prod_{i=1}^{L+1} \left\|T^{(i)}\right\|}{\left\|T^{(p)}\right\|} \leqslant$$

$$(C^2k^2w^2mn)^{2L} \left(\frac{1}{k_{\text{pool}}^2} \right)^{L-l+2-I\{p-1\leqslant l\}} \leqslant$$

$$(C^2k^2w^2mn)^{2L} \left(\frac{1}{k_{\text{pool}}^2} \right)^{L-l+2}.$$

Then we have

$$\|\mathbf{F}\|^2 \leqslant \|\mathbf{x}\|^2 (L+1)(k^2C^2w^2mn)^{(L)} \big(\frac{1}{k_{\text{pool}}^2}\big)^{L-l+2},$$

$$\|\mathbf{H}_O\| \leqslant \sqrt{2}\left\|\mathbf{x}^2\right\| q^2 \big(\frac{1}{k_{\text{pool}}^2}\big)^{L-l+2} (L+1)(k^2C^2w^2mn)^L,$$

where $q^2 = mnC^2k^2$. □

Appendix F
Proof of the Lemma 4

Proof. We use the notation $\mathbf{A}^{(l)}$ for our 2D-Avg-Pool layer One can see, that

$$(\mathbf{A}_{*,i}, \mathbf{A}_{*,j}) = \frac{1}{k_{\text{pool}}^4} I\{\text{i, j corresponds to the same window.}\}$$

To achieve this, we can see into formula:
$(\mathbf{A}_{*j}, \mathbf{A}_{*i}) = \sum_k \mathbf{A}_{ki}\mathbf{A}_{kj} = \sum_{k: \mathbf{A}_{ki}\neq 0, \mathbf{A}_{kj}\neq 0} \frac{1}{k_{\text{pool}}^4}$.
After this, applying elementary transformations over rows and columns, we reduce matrix $\mathbf{A}^{(p)^\intercal}\mathbf{A}^{(p)}$ to a block-diagonal form, where blocks correspons indexes in the same avg-pool window. Each block of $\mathbf{A}^{(p)^\intercal}\mathbf{A}^{(p)}$ is $\mathbf{B}_i = \frac{1}{k_{\text{pool}}^2}\mathbf{1}\mathbf{1}^\intercal$, where $\mathbf{1} = \mathbf{1}_{k_{\text{pool}}^2} \in \mathbb{R}^{k_{\mathbf{A}}^2}$ - vector of ones, and it's norm $\|\mathbf{B}_i\| = \frac{1}{k_{\text{pool}}^2}\left\|\mathbf{1}\mathbf{1}^\intercal\right\| = \frac{1}{k_{\text{pool}}}$
Norm of block-diagonal matrix (and the norm of a matrix that can be reduced to this form) is max of norms:
$\left\|\mathbf{A}^{(p)}\right\| = \max_i \|\mathbf{B}_i\| = \frac{1}{k_{\text{pool}}}$,
because of $\left\|\mathbf{A}^{(p)}\right\| \leqslant 1$, we can use completely reuse the calculations of the previous proof and have the same result. □

Appendix G
Proof of Lemma 5

Proof. As in previous proofs we need to estimate $\left\|\mathbf{G}^{(p)}\right\|^2 \left\|\mathbf{R}^{(p-1)}\right\|^2$.
We know, that $\left\|T^{(L+1+p)}\right\|^2 \leqslant (h^2\tilde{w}^2)\ \forall p = 1, \ldots, P$ Then, we can estimate

$$\left\|\mathbf{G}^{(p)}\right\|^2 \left\|\mathbf{R}^{(p-1)}\right\|^2 \leqslant (h^2\tilde{w}^2)^P (k^2C^2w^2mn)^L,$$

for $p \leqslant L+1$ and

$$\left\|\mathbf{G}^{(p)}\right\|^2 \left\|\mathbf{R}^{(p-1)}\right\|^2 \leqslant (h^2\tilde{w}^2)^{P-1} (k^2C^2w^2mn)^{L+1},$$

for $p = L+2 \ldots L+P+1$.
Or in one entry:

$$\left\|\mathbf{G}^{(p)}\right\|^2 \left\|\mathbf{R}^{(p-1)}\right\|^2 \leqslant (h^2\tilde{w}^2)^{P-I_{\{p>L+1\}}} (k^2C^2w^2mn)^{L+I_{\{p>L+1\}}}.$$

Then we have:

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

$$(h^2\tilde{w}^2)^P (k^2C^2w^2mn)^L \big(L+1+P\frac{h^2\tilde{w}^2}{k^2C^2w^2mn}\big).$$

And applying this result to the Hessian

$$\|\mathbf{H}_O\| \leqslant \sqrt{2}W_x q^2 (h^2\tilde{w}^2)^P (k^2C^2w^2mn)^L \times$$

$$\times \big(L+1+P\frac{h^2\tilde{w}^2}{k^2C^2w^2mn}\big).$$

□